



**PRACTICAL IMPLEMENTATION OF  
MULTIPLE MODEL ADAPTIVE ESTIMATION USING  
NEYMAN-PEARSON BASED HYPOTHESIS TESTING AND  
SPECTRAL ESTIMATION TOOLS**

DISSERTATION

Peter D. Hanlon  
Captain, USAF

AFIT/DS/ENG/96-07

**DISTRIBUTION STATEMENT A**

Approved for public release;  
Distribution Unlimited

DTIC QUALITY INSPECTED 2

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY  
**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

19970501 167

**AFIT/DS/ENG/96-07**

**PRACTICAL IMPLEMENTATION OF  
MULTIPLE MODEL ADAPTIVE ESTIMATION USING  
NEYMAN-PEARSON BASED HYPOTHESIS TESTING AND  
SPECTRAL ESTIMATION TOOLS**

**DISSERTATION**

**Peter D. Hanlon  
Captain, USAF**

**AFIT/DS/ENG/96-07**

Approved for public release; distribution unlimited

**AFIT/DS/ENG/96-07**

**PRACTICAL IMPLEMENTATION OF  
MULTIPLE MODEL ADAPTIVE ESTIMATION USING  
NEYMAN-PEARSON BASED HYPOTHESIS TESTING AND  
SPECTRAL ESTIMATION TOOLS**

**DISSERTATION**

Presented to the Faculty of the School of Engineering  
of the Air Force Institute of Technology  
Air University in Partial Fulfillment of the  
Requirements for the Degree of  
Doctor of Philosophy

Peter D. Hanlon, B.S., M.S.

Captain, USAF

September 1996

Approved for Public Release; distribution unlimited

**PRACTICAL IMPLEMENTATION OF  
MULTIPLE MODEL ADAPTIVE ESTIMATION USING  
NEYMAN-PEARSON BASED HYPOTHESIS TESTING AND  
SPECTRAL ESTIMATION TOOLS**

Peter D. Hanlon, B.S., M.S.  
Captain, USAF

Approved:

Peter S. Maybeck 11 Sep 96  
Peter S. Maybeck

M. Pachter 11 Sep 96  
Meir Pachter

Lang Hong 11 Sep 96  
Lang Hong

William E. Wiesel 11 Sep 96  
William E. Wiesel

Accepted: Robert A. Calico  
Dean, School of Engineering

## Acknowledgments

I am deeply indebted to many people for the successful completion of this dissertation. First to my research advisor, Dr. Peter Maybeck, for the encouragement, motivation, boundless energy, and greatly needed guidance throughout my internment at AFIT. I am particularly grateful for the understanding he showed when life was overwhelming.

Finally, words cannot express my gratitude to my wife, family, and church for the support they provided. My church provided the peaceful office that I needed to complete this dissertation, along with activities to help balance life. My kids gave me countless hours of uninterrupted computer time, believe me this was quite a sacrifice. Lastly, my wife patiently listened (sometimes with glazed eyes) to my rambling about my latest difficulties with math, learned to deal with an exhausted husband, and dealt with the kids when dad was staring at the computer screen. For this and much more, thanks Kathy. I love you.

Peter D. Hanlon

## Table of Contents

	Page
Acknowledgments .....	iii
List of Figures .....	vii
List of Tables .....	x
Abstract .....	xi
I. Introduction .....	1
1.1 Chapter Overview .....	1
1.2 Description of the Motivating Problem .....	1
1.3 Dissertation Overview .....	8
II. Background .....	9
2.1 Chapter Overview .....	9
2.2 Multiple Model Adaptive Estimation .....	9
2.2.1 Multiple Model Adaptive Estimation Development .....	9
2.2.2 Multiple Model Adaptive Control Development .....	10
2.2.3 Multiple Model Adaptive Estimation Based Control .....	12
2.2.4 Multiple Model Adaptation Applications .....	14
2.3 Signal Processing Techniques .....	14
2.3.1 Spectral Estimation Techniques .....	14
2.3.2 Neyman-Pearson Hypothesis Testing .....	15
2.4 Chapter Summary .....	17
III. Theory Development .....	18
3.1 Chapter Overview .....	18
3.2 Kalman Filter Bank .....	19
3.2.1 Basic Equations .....	19
3.2.2 Nomenclature for Representing Mismatching .....	24
3.2.3 Covariance of the Residual .....	29
3.2.4 Mean of the Residual .....	32
3.2.4.1 No Mismatching .....	36
3.2.4.2 Mismatched Input Matrix .....	36
3.2.4.3 Mismatched Output Matrix .....	37
3.2.4.4 Mismatched State Transition Matrix .....	38
3.2.5 Kalman Filter Bank Implemented Using a Single Residual .....	38
3.2.5.1 Model Difference Nomenclature .....	41
3.2.5.2 Computation of the Residuals .....	42
3.2.5.3 Different Input Matrix Models .....	45
3.2.5.4 Different Output Matrix Models .....	46
3.2.5.5 Different State Transition Matrix Models .....	49

	Page
3.2.6 Residual Correlation Kalman Filter Bank .....	50
3.2.6.1 Basic Equations .....	53
3.2.6.2 Mean of the Power Spectral Density Estimate .....	56
3.2.6.3 Covariance of the Power Spectral Density Estimate .....	63
3.3 Hypothesis Testing Algorithm .....	66
3.3.1 Standard Hypothesis Testing Algorithm .....	68
3.3.1.1 Basic Equations .....	69
3.3.1.2 $\beta$ Stripping .....	73
3.3.1.3 Lower Bounding of the Conditional Probabilities .....	74
3.3.1.4 Filter Tuning .....	75
3.3.1.5 Smoothing of the Conditional Probabilities .....	76
3.3.1.6 Exponential Penalty Increase .....	76
3.3.1.7 Propagating Several Time Samples before Updating .....	77
3.3.2 Neyman-Pearson based Hypothesis Testing Algorithm .....	77
3.3.2.1 Basic Equations .....	77
3.3.2.2 Single Time Sample Hypothesis Testing .....	82
3.3.2.3 Multiple Time Samples Hypothesis Testing .....	90
3.4 Chapter Summary .....	100
IV. Results .....	102
4.1 Chapter Overview .....	102
4.2 Characterization of the Residual .....	103
4.2.1 No Mismatching (No Failure) .....	105
4.2.2 Mismatched Input Control Matrix (Actuator Failure) .....	107
4.2.3 Mismatched Output Matrix (Sensor Failure) .....	117
4.3 Single Residual Kalman Filter Bank .....	122
4.3.1 No Mismatching (No Failure) .....	122
4.3.2 Mismatched Control Input Matrix (Actuator Failure) .....	126
4.3.3 Mismatched Output Matrix (Sensor Failure) .....	130
4.3.4 Overall Equivalency .....	132
4.3.5 Generalized Likelihood Ratio Class Equivalence .....	132
4.4 Failure Detection Performance .....	134
4.4.1 SKFB-SHTA Failure Identification Performance .....	139
4.4.2 RCKFB-SHTA Failure Identification Performance .....	155
4.4.3 SKFB-NPHTA Failure Identification Performance .....	160
4.4.4 Comparative Failure Identification Performance .....	165
4.5 Chapter Summary .....	168
V. Conclusions .....	170
5.1 Chapter Overview .....	170
5.2 Characterization of the Residual .....	170
5.3 Single Residual Kalman Filter Bank .....	171
5.4 Residual Correlation Kalman Filter Bank .....	172
5.5 Neyman-Pearson Hypothesis Testing Algorithm .....	173

	Page
5.6 Recommendations for Future Research .....	174
Bibliography .....	177
Vita .....	183

## List of Figures

Figure	Page
1. Multiple Model Adaptive Estimation Algorithm .....	2
2. LAMBDA Flight Envelope .....	4
3. Multiple Model Adaptive Control Algorithm .....	11
4. Multiple Model Adaptive Estimation Algorithm .....	18
5. Alternative Method of Computing a Second Kalman Filter Residual .....	39
6. Alternative Multiple Model Adaptive Estimation Filter Bank Structure Using Equivalent Residuals .....	40
7. Residual from the Kalman Filter with the Fully Functional Model in the Presence of an Elevator Failure .....	51
8. Fourier Transform of the Residual from the Fully Functional Kalman Filter with Mismatching (solid line) and No Mismatching (dotted line) .....	52
9. Multiple Model Adaptive Estimation Algorithm using a Residual Correlation Kalman Filter Bank .....	55
10. Comparison of Updated State Estimates With Nonupdated State Estimates .....	78
11. Implementation of the Neyman-Pearson Based Hypothesis Test .....	85
12. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with No Mismatching .....	106
13. Control Inputs, Potted in Radians .....	109
14. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismatched Elevator Input. ....	110
15. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismatched Aileron Input. ....	114
16. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismatched Rudder Input. ....	116
17. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismatched Forward Velocity Sensor. ....	118

18. Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Pitch Rate Sensor .....	121
19. The Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Kalman Filter Residual for a No Failure Test Case .....	123
20. The Left Elevator Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case .....	125
21. The Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Kalman Filter Residual for a Left Elevator Failure Test Case .....	127
22. The Left Aileron Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case .....	129
23. The Pitch Rate Failure Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Pitch Angle Sensor Failure Test Case. ....	131
24. Multiple Model Adaptive Estimation Algorithm .....	134
25. Alternative Multiple Model Adaptive Estimation Filter Bank Structure Using Equivalent Residuals .....	135
26. Multiple Model Adaptive Estimation Algorithm using a Residual Correlation Kalman Filter Bank .....	136
27. Equivalent Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm Structure .....	137
28. Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Left Elevator Failure .....	140
29. Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Left Rudder Failure .....	141
30. Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Forward Velocity Sensor Failure .....	142
31. Overall Failure Identification Performance of a Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) .....	144

32. Actuator Failure Identification Performance for the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) .....	146
33. Kalman Filter Likelihood Quotients for the No-Failure Case .....	149
34. Kalman Filter Likelihood Quotients for the Right Rudder Failure Case .....	150
35. Kalman Filter Likelihood Quotients for a Right Rudder Failure Injected at 4.2 Seconds ....	151
36. Elevator Failure Identification Performance for Various Failure Injection Times .....	153
37. Aileron Failure Identification Performance for Various Failure Injection Times .....	153
38. Rudder Failure Identification Performance for Various Failure Injection Times .....	154
39. Actuator Failure Identification Performance of a Residual Correlation Kalman Filter Bank (RCKFB) with a Standard Hypothesis Testing Algorithm (SHTA) .....	156
40. Actuator Failure Identification Performance for the Residual Correlation Kalman Filter Bank (RCKFB) with a Standard Hypothesis Testing Algorithm (SHTA) .....	159
41. Overall Failure Identification Performance of a Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA) .....	161
42. Actuator Failure Identification Performance for the Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA) .....	164
43. Comparative Failure Identification Performance for an Elevator Failure. ....	166
44. Comparative Failure Identification Performance for an Aileron Failure. ....	166
45. Comparative Failure Identification Performance for a Rudder Failure. ....	167

## List of Tables

Table	Page
1. Kalman Filter Residual Statistical Comparison when No Mismodeling Exists. ....	107
2. Kalman Filter Residual Statistical Comparison for the Case of an Elevator Failure .....	112
3. Kalman Filter Residual Statistical Comparison for the Case of an Aileron Failure .....	115
4. Kalman Filter Residual Statistical Comparison for the Case of a Rudder Failure .....	115
5. Kalman Filter Residual Statistical Comparison for the Case of a Forward Velocity Sensor Failure. ....	119
6. Kalman Filter Residual Statistical Comparison for the Case of a Pitch Rate Sensor Failure. .	120
7. Temporal Average of the Difference Between the Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Filter Residual for a No-Failure Test Case. ....	124
8. Temporal Average of the Difference Between the Left Elevator Failure Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case. ....	126
9. Temporal Average of the Difference Between the Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Filter Residual for a Left Elevator Failure Test Case. ....	128
10. Temporal Average of the Difference Between the Left Aileron Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case. ....	130
11. Temporal Average of the Difference Between the Pitch Rate Failure Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Pitch Angle Failure Test Case. ....	132
12. Overall Temporal Average of the Difference Between an Actual Residual and the Equivalent Residual, Average over all 4096 Test Cases. ....	133
13. SKFB-SHTA Actuator Failure Identification Performance, 10-Test Sample Average. ....	146
14. RCKFB-SHTA Actuator Failure Identification Performance, 4-Test Sample Average. ....	159
15. SKFB-NPHTA Actuator Failure Identification Performance, 10-Test Sample Average ....	164

**ABSTRACT**

This study investigates and develops various modifications to the Multiple Model Adaptive Estimation (MMAE) algorithm. The standard MMAE uses a bank of Kalman filters, each based on a different model of the system. Each of the filters predict the system response, based on its system model, to a given input and form the residual difference between the prediction and sensor measurements of the system response. Model differences in the input matrix, output matrix, and state transition matrix, which respectively correspond to an actuator failure, sensor failure, and an incorrectly modeled flight condition for a flight control failure application, were investigated in this research. An alternative filter bank structure is developed that uses a linear transform on the residual from a single Kalman filter to produce the equivalent residuals of the other Kalman filters in the standard MMAE. A Neyman-Pearson based hypothesis testing algorithm is developed that results in significant improvement in failure detection performance when compared to the standard hypothesis testing algorithm. Hypothesis testing using spectral estimation techniques is also developed which provides superior failure identification performance at extremely small input levels.

# **PRACTICAL IMPLEMENTATION OF MULTIPLE MODEL ADAPTIVE ESTIMATION USING NEYMAN-PEARSON BASED HYPOTHESIS TESTING AND SPECTRAL ESTIMATION TOOLS**

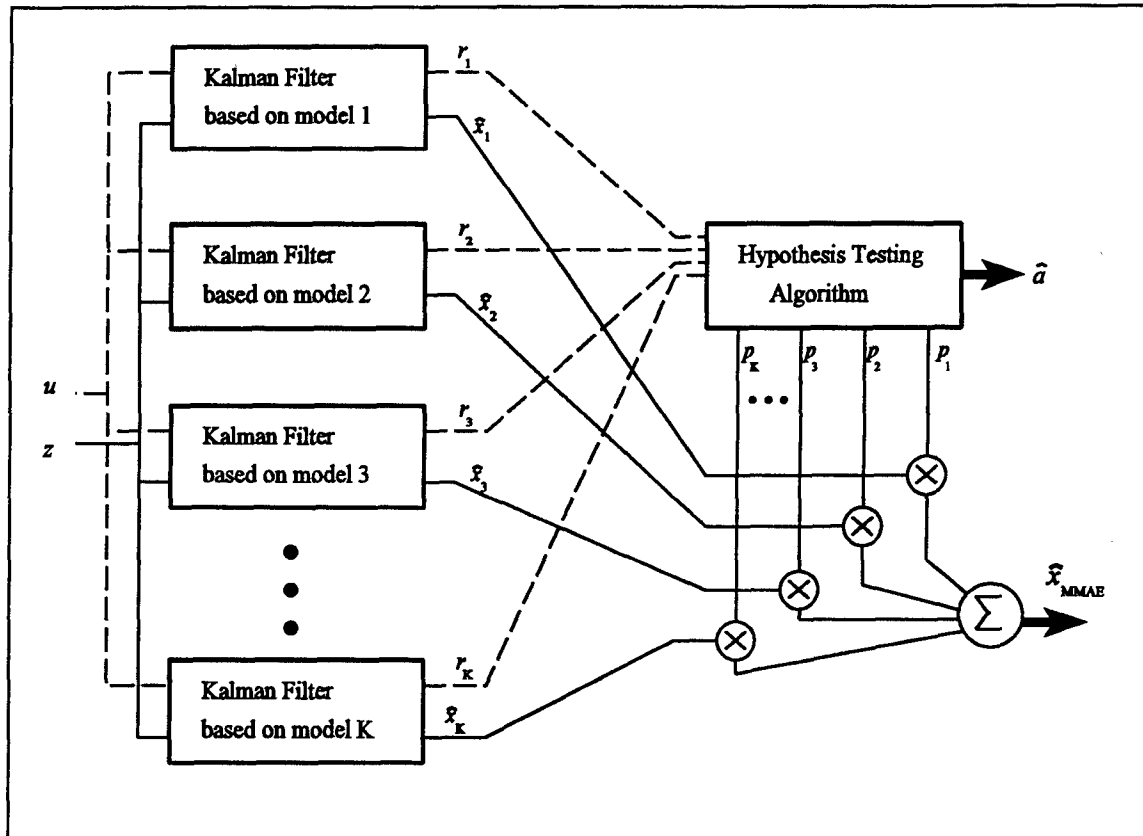
## **I. Introduction**

### **1.1 Chapter Overview**

In this chapter we introduce the development of modifications to the Multiple Model Adaptive Estimation (MMAE) algorithm. To demonstrate the MMAE performance with these modifications, we have chosen to use a flight control failure detection application, which is described in Section 1.2. We also will point out some of the difficulties encountered with contemporary MMAE designs that motivated the development of these modifications. Finally, in Section 1.3 we present an overview of the structure of this dissertation.

### **1.2 Description of the Motivating Problem**

The MMAE algorithm shows tremendous promise in enhancing flight control performance. The constant pressure to improve aircraft performance, particularly in military aircraft, has forced designers to build aircraft with instabilities that are beyond that capabilities of human pilots to control. Thus, these designers have been forced to rely on increasingly complex flight control systems. These systems are designed for a particular system configuration (i.e. a fully functional



**Figure 1.** Multiple Model Adaptive Estimation Algorithm

aircraft) and usually cannot perform adequately when the system is configured substantially differently (i.e. a flight control failure). Designers are constructing several flight control system designs that will perform adequately for different system configurations, but the flight control system must be informed which configuration is the appropriate one to choose. One such method of detecting and identifying system failures is the MMAE.

The MMAE, diagrammatically shown in Figure 1, is composed of a bank of Kalman filters, each imbued with its own model, and a hypothesis testing algorithm. Each elemental Kalman filter uses its own model, along with a given input ( $u$ ), to develop an estimation of the current aircraft states ( $\hat{x}_t$ ), independent of the other filters. The filter then uses this estimate, along with the current

measurement of those states ( $z$ ), to form the residual ( $r_k$ ), which is the difference between the measurement and the filter's prediction of the measurements before they arrive. The residuals from the filters are used by the hypothesis testing algorithm as a relative indication of how close each of the filter models are to the true model, i.e., to the real-world situation. The smaller the residual, the closer the filter model matches the true model. The hypothesis testing algorithm first scales the residuals to account for various uncertainties and noises in the measurements (as developed in detail in Section 3.3.1), and then computes the conditional probability for each of the hypotheses modeled in the bank of Kalman filters ( $p_k$ ). These probabilities are then used to weight the individual Kalman filter state estimates to produce a blended estimate of the true states ( $\hat{x}_{MMAE}$ ), which can then be used as the optimal estimate of the states by a control system. When used for failure identification, each of the Kalman filters would model a different failure condition and the residuals from each filter would indicate how close that filter's model is to the actual failure condition. By monitoring these residuals, the hypothesis tester can estimate the current failure status of the aircraft ( $\hat{a}$ ).

The specific aircraft model [62] that is used for this application was developed for the LAMBDA flight vehicle. The LAMBDA is an unmanned research vehicle developed by the Flight Control Division of the Flight Dynamics Directorate, Wright Laboratory, as an affordable, flexible research vehicle for testing and demonstrating flight control concepts, devices, and systems [62]. The flight condition of the LAMBDA is determined by five parameters: the aircraft weight, forward velocity, dynamic pressure, center of gravity, and trim angle. The average flight condition parameters are a weight of 200 lbs, a speed of 160 ft/sec, a dynamic pressure of 30.43 lb/ft<sup>2</sup>, a center of gravity located at 46.8 inches from the aircraft nose, and a trim angle of zero degrees. The LAMBDA usually flies at low altitude; therefore the dynamic pressure ( $\frac{1}{2}\rho v^2$ ) is directly related to forward velocity ( $v$ ) since the air density ( $\rho$ ) is essentially constant. The trim angle is usually very

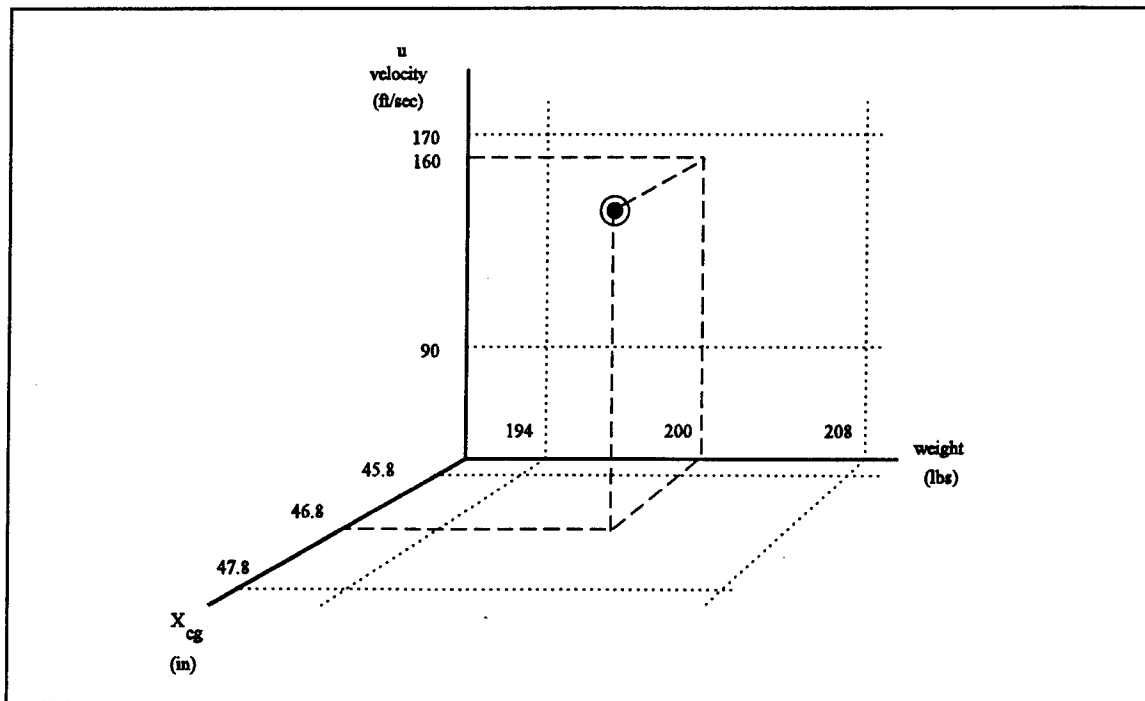


Figure 2. LAMBDA Flight Envelope

small for a well trimmed aircraft; therefore we assume a trim angle of zero. Since the altitude and trim angle are considered constant, we end up with a three-dimensional flight envelope for the LAMBDA as shown in Figure 2, which also shows the design flight condition. This flight condition is considered average because the LAMBDA is usually flown close to maximum velocity, at low altitude, with the variation in weight primarily dependent on the amount of fuel consumed. Therefore the design flight condition is a median value of the normal operating points.

Embedded within the linearized model (linear perturbation model about trim condition) of the LAMBDA are several other assumptions that were used to develop the model [62], namely lateral and longitudinal axis decoupling and first order actuator dynamics. The linearized model was developed by estimating flight coefficients from geometric data and then adjusting these estimates using good engineering judgment and flight test data. For a fully functional aircraft, the cross-axis

coupling terms are so small that they produce a negligible aircraft response. Currently, all test flights have been conducted using a fully functional aircraft [62], and therefore, the flight test data does not contain enough information to estimate the cross-axis terms. The assumption of first order actuator dynamics compares quite well with the flight test data.

To model single actuator failures, we have assumed that a single flight control surface failure will produce half of the expected response from dual control surfaces. For instance, a certain elevator input might produce a 10 degree pitch-up response. We have assumed that, for this same input, a failure of only the right elevator actuator would produce half of this response (5 degree pitch-up) since only the left elevator actuator would be functional, thereby decreasing the actuator surface area by one half.

As mentioned above, we have assumed that the cross-axis coupling terms are negligible even for single surface failures, primarily because there is insufficient data to estimate these terms [62]. Using our example, a right elevator failure would produce a small yawing and rolling response, which we assume to be negligible. An exception to this assumption is the cross-coupling between the ailerons and the yaw axis because the long yaw moment arm of the ailerons could easily produce a significant yaw. Fortunately, there is sufficient data to estimate the aileron's effect on the yaw rate ( $r$ ) and the rudder's effect on the roll rate ( $p$ ). At the design point, the contribution of the aileron to a yaw rate is 10.5% and the contribution of the rudder to a roll rate is only 0.3%. This supports our assumption that the cross-axis terms are small. However, Swift found noticeable roll/pitch coupling dynamics for the LAMBDA, but was unable to estimate the cross coupling terms [62: 3.50]. A flight test with single actuator failures should provide the much needed data for estimating these cross-axis coupling terms, and the aircraft model can then be corrected for further tests.

Previous research [19, 31, 41] that developed an MMAE-based failure detection algorithm for this flight vehicle identified three problems with contemporary MMAE designs. These problems are a lack of a methodical way to make a definitive failure status declaration, the computational cost for implementing these designs, and the need to dither the system constantly to provide adequate excitation for good failure detection. These problems are described in more detail below.

Most flight control systems require a definite failure status declaration from the failure detection and identification algorithm. The conditional probabilities produced by the MMAE give only a relative indication of the failure status. To make a declaration of the failure status, contemporary MMAE designers choose a probability threshold that is used to declare if a certain hypothesized failure has occurred. If the conditional probability for that hypothesis exceeds the threshold, then the hypothesized failure would be declared. Unfortunately, there is no method of choosing this threshold aside from extensive computer simulations. These simulations would give a relative indication of the tradeoff between the false alarm rate and failure detection performance and the designer would use good engineering judgement to choose the threshold. A more satisfactory design method would be to be able to dictate the maximum acceptable false alarm rate and desired failure detection probability, and have these two parameters dictate the threshold. Such a technique is available using a Neyman-Pearson base hypothesis testing algorithm that will be developed in this research.

MMAE can be a very costly algorithm to implement. The Kalman filter algorithm, presented in Section 3.2.1, can be computationally intensive, particularly if a large number of filter states is needed to capture the real-world aircraft dynamics. Using a bank of Kalman filters further compounds the computational cost of implementing an MMAE algorithm. MMAE can be implemented on a set of parallel processors, because of its inherent parallel structure, but the

required power, cooling, and data bus interconnection, along with the weight and cost, of producing this set of parallel processors could easily make this implementation infeasible. We develop an equivalent MMAE filter bank structure that uses the residual and state estimates from a single Kalman filter, along with the linear transformation that is developed in Section 3.2.5, to produce the equivalent residual from another Kalman filter. This transformation computes the equivalent residual by using the known differences between the two Kalman filter models. These model differences usually produce sparse matrices, which significantly reduces the computational cost of computing the equivalent residuals. Thus, the bank of filters can be replaced by a single filter with several linear transformations, each of which produces the equivalent residual to the Kalman filter that it replaces, but with a reduction in the cost of computing the residual.

The primary objection to implementing an MMAE-based failure detection algorithm is the need to dither the system constantly. The MMAE compares the magnitudes of the residuals (appropriately scaled to account for various uncertainties and noises) from the various filters and chooses the hypothesis that corresponds to the residual that has a history of having smallest (scaled) magnitude. Large residuals need to be produced by the filters with models that are incorrect to be able to identify these incorrect hypotheses. The residual is the difference between the measurement of the system output and the filter's prediction of what that measurement should be, based on the filter-assumed system model. Therefore, to produce the needed large residuals in the incorrect filters, we need to produce a history of sufficiently large system outputs, so we need to dither the system constantly and thereby excite the system states. For flight control failure applications, we would need to move the aircraft continually in all axes to produce the desired failure detection performance, to which most pilots (not to mention passengers) would strenuously object. We develop a modified MMAE algorithm that uses the correlation of the residual with signal processing techniques to

produce the desired failure detection, while significantly reducing the objectionable dither, possibly to the point of being subliminal.

### 1.3 Dissertation Overview

In the subsequent chapters we develop this research in more detail. We have presented a general introduction to the MMAE and a brief description of the problems that have motivated this research. Chapter 2 gives a brief background on the development of the MMAE and a description of some statistical signal processing techniques that are used in the research. Chapter 3 presents the development of the theory behind the various modifications to the MMAE that have been developed. Chapter 4 presents the results from a specific simulation of these modifications using the flight control application described above. Finally, in Chapter 5 we present a comparison of the various modifications, draw pertinent conclusions, and finish with our recommendations for future research.

## II. Background

### 2.1 Chapter Overview

In this chapter we introduce the background research that has led up to this investigation. We start, in Section 2.2, with a brief survey of the development of the Multiple Model Adaptive Estimation (MMAE) and describe some of the research of its use for various applications. In Section 2.3 we present an overview of statistical signal processing techniques that may help enhance the performance of MMAE. These techniques include hypothesis testing based on various spectral estimation techniques and the Neyman-Pearson lemma .

### 2.2 Multiple Model Adaptive Estimation

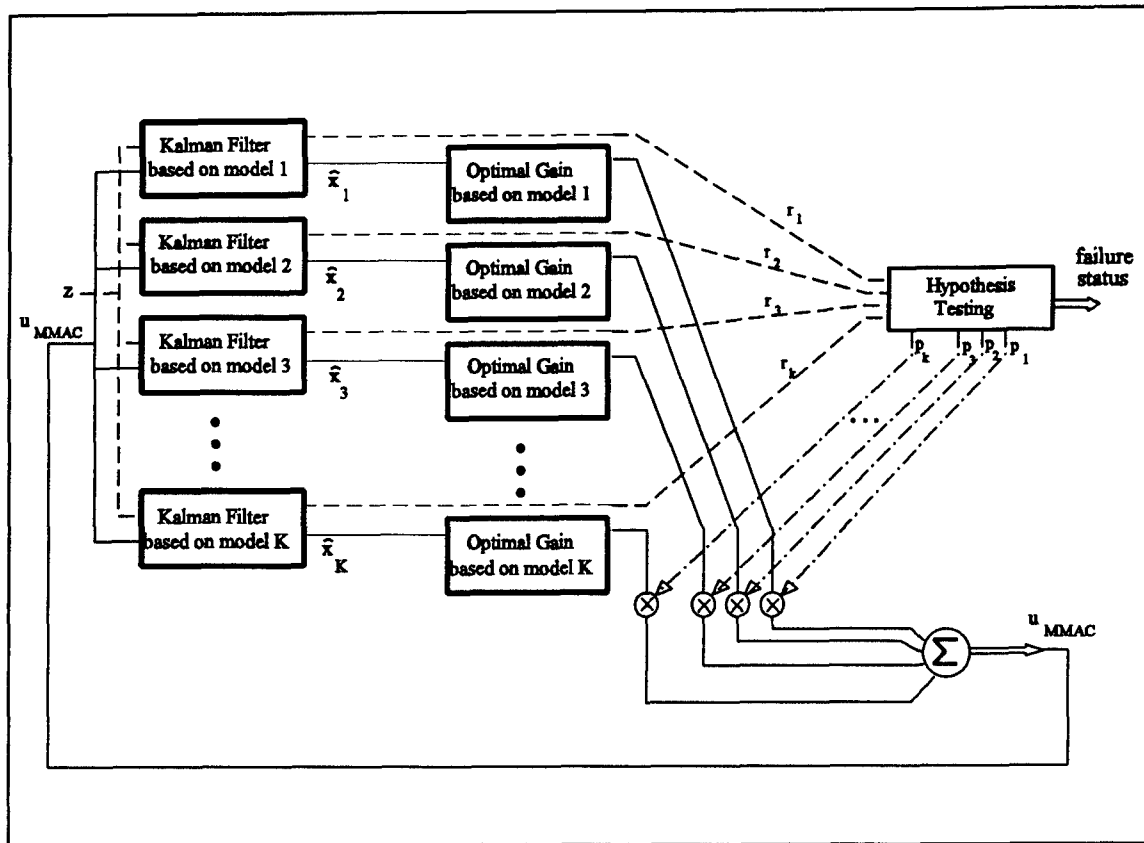
2.2.1 Multiple Model Adaptive Estimation Development. The use of multiple filters in a parallel structure to generate adaptive estimation algorithms was first developed by D. T. Magill [35]. He arranged a number of Kalman filters, each with different time invariant plant models, and used the residuals from these filters to form an appropriately weighted sum of the Kalman filter estimates, as shown in Figure 1. He showed that this adaptive estimation algorithm produced the optimal estimate in the minimum mean square error sense for a Gauss-Markov process, over a finite set of models, one of which is the "true" model which accurately represents the true system. This work was extended to handle equivalent discrete-time systems by C. B. Chang and M. Athans [6, 7, 8].

The properties of this multiple model adaptive estimation concept were further developed by Lainiotis and others [10, 20, 28, 29]. They investigated the properties of this concept when using a discrete number of models to represent a continuous domain of plant models [28, 29]. With others,

he showed that good performance was attained with tightly tuned Kalman filters, but these filters produced erratic behavior if the filter with the correct model was not in the filter bank [10, 20].

A. S. Willsky [68] surveyed a number of failure detection methods, using a performance index as an empirical measurement of the capabilities of the various methods. He surveyed three classes of failure detection methods. The first was the use of specific failure-sensitive filters, such as limited memory or Fagin age weighting filter and filters that included failure states in the filter model. A second class was methods that conduct statistical tests on the filter innovations, such as the Generalized Likelihood Ratio (GLR) and voting techniques. The third class consisted of multiple hypothesis filter detectors, specifically the MMAE. The performance index he used included the types of failure modes that can be detected, implementation complexity, various performance measures (false alarms, detection delays, repeatability), and robustness in the presence of modeling errors. He found that the multiple model adaptive estimator will yield the best performance over the widest class of failures.

2.2.2 Multiple Model Adaptive Control Development. M. Athans et. al., extended this adaptive estimation concept to adaptive control developed for the flight control system of NASA's F-8C flight test aircraft [1]. They weighted the optimal control signals generated by a bank of Linear system-Quadratic cost function-Gaussian noise distribution (LQG) controllers, each with an embedded Kalman filter using a different plant model, and then summed these weighted control signals together, as shown in Figure 3. They found that this algorithm provided good control at flight conditions that were close to the design conditions of the Kalman filter models, as long as the system was appropriately excited, using a test signal or dither, to attain good failure identification. They also found that this algorithm was sensitive to high frequency noise, specifically strong wind gusts, and required low pass filtering to attain good performance.



**Figure 3.** Multiple Model Adaptive Control Algorithm

Greene and Willsky [16] examined and defined stability regions where the multiple model adaptive control (MMAC) algorithm yields non-oscillatory responses. They found that the stability of the MMAC was determined by the relation between the growth rate of the most unstable mode of a Kalman filter with a mismatched model, when compared to the truth model, and the rate of decay of the slowest stable mode of a Kalman filter with a matching model. They presented a method of computing the borders where the algorithm is neutrally stable, which then defines the "domain of attraction," or the region where the MMAC's response does not oscillate.

Longitudinal control of the Short Take-Off and Landing (STOL) F-15 using Multiple Model Adaptive Control (MMAC) was investigated by Pogoda [42, 52] and Stevens [43, 60]. Pogoda

developed an algorithm for the landing phase of the flight regime, which would reconfigure the flight control in the presence of a single failed control surface or sensor. He designed Kalman filter models and controllers for a fully functional aircraft, a failed stabilator, a failed "pseudo-surface," and a failed pitch rate sensor. The "pseudo-surface" was a combination of the canards, ailerons, and flaps, to allow a reduction in the number of aircraft states and independent control surfaces. Stevens extended this research further by including failed reverser vanes, a failed velocity sensor, and a failed flight path sensor. He first investigated "soft" or partial failures of these sensors and surfaces, with the soft failures modeled as either partial power to a flight surface, an increase in the sensor noise, or an increase in the sensor bias. He then investigated the performance of a hierarchical structure of multiple model adaptive controllers to detect the presence of double failures, such as a stabilator surface failure followed by a flight path angle sensor failure. This hierarchical structure of controllers started with a bank of controllers, with different single failure models (including, of course, a no-failure model), that would detect the first failure and then switch to another bank of controllers that had models that assumed both the detected failure and a second failure (including no second failure). Each of the banks also had a controller with a failure model that assumed that the detected failure had not occurred after all, which allowed the structure to correct any misidentifications. Pogoda and Stevens both found that the MMAC structure was able to identify the failures properly and reconfigure the control law to maintain stable flight control. Stevens found that the MMAC would blend the appropriate controller commands in the presence of soft failures, and that the hierarchical structure would properly detect multiple failures.

**2.2.3 Multiple Model Adaptive Estimation Based Control.** In a similar manner, Stratton [61] and Menke [47, 48, 49] have developed an MMAE-based control algorithm for both the longitudinal and lateral axes of the VISTA F-16 aircraft. In contrast to the MMAC of

Figure 3, this algorithm uses the MMAE of Figure 1, feeding  $\hat{x}$  and  $\hat{x}_{MMAE}$  to a single controller block; in their implementation, only  $\hat{x}_{MMAE}$  was used as an input to an already designed and validated flight control system. Stratton used a flight condition of Mach 0.8 and altitude 10,000 feet, while Menke designed for a flight condition of Mach 0.4 and altitude 20,000 feet, the latter involving low dynamic pressures and thus presenting a more difficult failure detection problem. They investigated both single and multiple, hard and soft, actuator and sensor failures. Included in their study was the effect of a test signal or commanded dither to aid in identifying the failure during benign straight and level flight conditions. Several dither signals were tested, including sine waves, square waves, triangular waves, and pulse trains, at levels that were deemed either subliminal (up to  $\pm 0.1$  g's in the longitudinal axis and  $\pm 0.2$  g's in the lateral axis) or non-subliminal (reasonable physical acceleration limits at the pilot's station). They found that the MMAE identified these failures correctly as long as an appropriate dither signal was present that excited all failure modes in both axes.

Eide [11, 12] and Stepaniak [58, 59] continued the development of the MMAE-based control algorithm for the VISTA F-16. Eide tested the algorithm against a full-scale nonlinear truth model, instead of a linearized truth model that was used for the previous performance analyses. Initially, he found fairly poor failure identification performance that was attributed to a mismatch between the full-scale nonlinear model and the linearized design model. He tuned the Kalman filters and achieved much better performance, thus showing the robustness of the MMAE algorithm, even when operating with a nonlinear system while the MMAE Kalman filters are based on linearized models. Stepaniak used an MMAE to redistribute the control input to redundant nonfailed actuators using the existing VISTA F-16 control system. He obtained excellent results, showing that this MMAE-based control algorithm could track a desired state trajectory, as long as the redundant actuators were not forced into saturation. Thus, he showed that the MMAE-based control algorithm

can completely compensate for a failed actuator, up to the limits of command authority, if redundant control surfaces are available.

2.2.4 Multiple Model Adaptation Applications. Multiple model adaptive algorithms have been successfully developed for a number of other applications. It was investigated for the detection and tracking of maneuvering targets [8, 14, 15, 37, 44, 50, 51, 63, 64, 65], flexible space structure control [13, 23, 24, 32, 33, 54], multiple hypotheses testing [1, 45], and prevention of the initial divergence of extended Kalman filters due to large initial uncertainties [22, 46]. It has been studied for use in diverse applications such as instrument failure detection in a pressurized water fusion reactor [10], autonomous monitoring of cardiac patients [17], adaptive signal processing of seismic data [58], and detection of incidents on freeways [69].

## 2.3 Signal Processing Techniques

2.3.1 Spectral Estimation Techniques. Kay [26] presents a methodical survey of some of the many spectral estimation techniques. He groups these techniques into classical methods, those that are based on Fourier transform and filtering theory, and modern techniques that are based on time series analysis and filtering theory. In the classical group he describes periodograms, averaged periodograms, and Blackman-Tukey spectral estimators. Based on the definition of the power spectral density of a signal, the Fourier transform of the autocorrelation function, these techniques window the data (using various shaped windows for each of the different techniques) and then compute the spectral content across the data window at various frequencies by computing the Fourier transform of an estimate of the autocorrelation function of the windowed data. Modern techniques are based on parametric modeling of the signal, with the assumption that the signal is composed of a known number of sinusoids.

The primary motivation for the research of these techniques is detecting and characterizing sinusoids with time-varying amplitude, embedded in a background of noise, which describes many applications where the data set is relatively short compared to the duration of the sinusoid. This occurs with speech synthesis, where a speech sound may only last for about 20 to 80 msec [36], and in Doppler radar and sonar, where the propagation characteristics of the transition medium may change in time [25]. In other applications the small data set is due to a genuine lack of data. Seismic data sets are transient because events such as volcanic eruptions and earthquakes last for very short periods of time [30]. Other applications have small data sets because of the prohibitive cost of collecting the data, such as optical interferometry [4], vibration analysis [2], radio astronomy [67], image processing [18], and the application that we are studying, flight control failure detection. There are many other methods of signal analysis that apply, such as Maximum Entropy Methods (MEM), particularly useful for small data sets [53], but we have chosen to research the use of periodograms (presented in Section 3.2.6) because it has been extensively researched, is straightforward to implement, and provides a basic conceptual framework for implementing the other techniques.

2.3.2 Neyman-Pearson Hypothesis Testing. Scharf [53:103-166] presents the development of a Neyman-Pearson Detector. The basis of the detector is a hypothesis test in which the distribution of a test statistic is known, for each of the hypotheses that is being tested. The hypothesis test is based on the Neyman-Pearson lemma, which will be formally presented in Section 3.3.2, which shows that a likelihood ratio that is formed using these known distributions of the test statistic, provides the best probability of detection for a given false alarm probability, for a fixed number of data samples. The probability of detection is defined as the probability that the correct hypothesis is actually chosen, given that the incorrect hypothesis is considered to be in force at the

time of the hypothesis test. For example, the probability of detection is the probability that, if a failure occurs, the hypothesis test chooses the failure hypothesis when prior to the test the no failure hypothesis was correct. The false alarm probability is defined as the probability that the incorrect hypothesis is chosen, given that the correct hypothesis is considered in force at the time of the hypothesis test. For example, this is the probability that, if a failure does not occur, the hypothesis test chooses the failure hypothesis, when prior to the test the no failure hypothesis was correct. Thus, we want to maximize the probability of detection, while minimizing the probability of false alarm, which illustrates the importance of the Neyman-Pearson lemma.

These definitions will differ slightly from the usual convention when the hypothesis that is considered in force is not the fully functional hypothesis. To illustrate this, assume that the left elevator has failed and that hypothesis is in force. If a hypothesis test is conducted, and the test incorrectly chooses the fully functional hypothesis, that would be a false alarm by the definitions given above. The usual convention is to call this incorrect declaration a missed detection because the test mistakenly missed the detection of the failed elevator. We will use the definitions given above to be consistent with the development of the Neyman-Pearson hypothesis test.

Frequently, the fixed length of the data set will not provide enough information to be able to distinguish the various hypotheses from each other. If the number of data samples is not set, thus defining a sequential estimation problem, the sequential probability ratio test (SPRT) is superior to the Neyman-Pearson most powerful test because it will not make a decision until sufficient information is obtained to distinguish the hypotheses from each other [34]. We have chosen to use the Neyman-Pearson most powerful test to lay the conceptual framework for this type of hypothesis testing. Scharf [53] also develops the methodology for selecting the decision threshold that is used to choose between the various hypotheses. The special case of a test statistic that is normally

distributed with differing means is particularly important for our application, since the Kalman filter residual will be shown, in Section 3.2, to be normally distributed with a nonzero mean if the Kalman filter model is inaccurate, and thus we can use the residual as a test statistic.

## 2.4 Chapter Summary

In this chapter we have presented a brief summary of the research that has led up to this investigation. First, we described the development of the MMAE, along with two methods of using the multiple model concept for control applications. The MMAC algorithm uses a bank of the Kalman filters with LQG controllers that were each developed for a specific hypothesis, and then uses the conditional probabilities from the hypothesis testing algorithm, to weight the commanded inputs from the various controllers. The MMAE-based control algorithm uses an MMAE to provide estimates, rather than raw measurements, to a single separate control system. The MMAE-based control algorithm with control redistribution has been shown to be able to compensate completely for any failed aircraft sensor and for any failed actuator, up to the point of saturation of the actuator, if redundant control surfaces are available. We briefly listed some of the applications that have used MMAE. Then, we described some spectral estimation techniques which show great promise in enhancing the failure detection performance of the MMAE. We have chosen to focus on the periodogram spectral estimation technique because its characteristics are well researched and it provides an excellent conceptual framework for developing other spectral estimation techniques. We introduced the Neyman-Pearson detector, which we will use as a framework for developing an alternative hypothesis testing algorithm.

### III. Theory Development

#### 3.1 Chapter Overview

A Multiple Model Adaptive Estimator (MMAE) consists of a bank of parallel Kalman filters, each with a different model, and a hypothesis testing algorithm as shown in Figure 4.

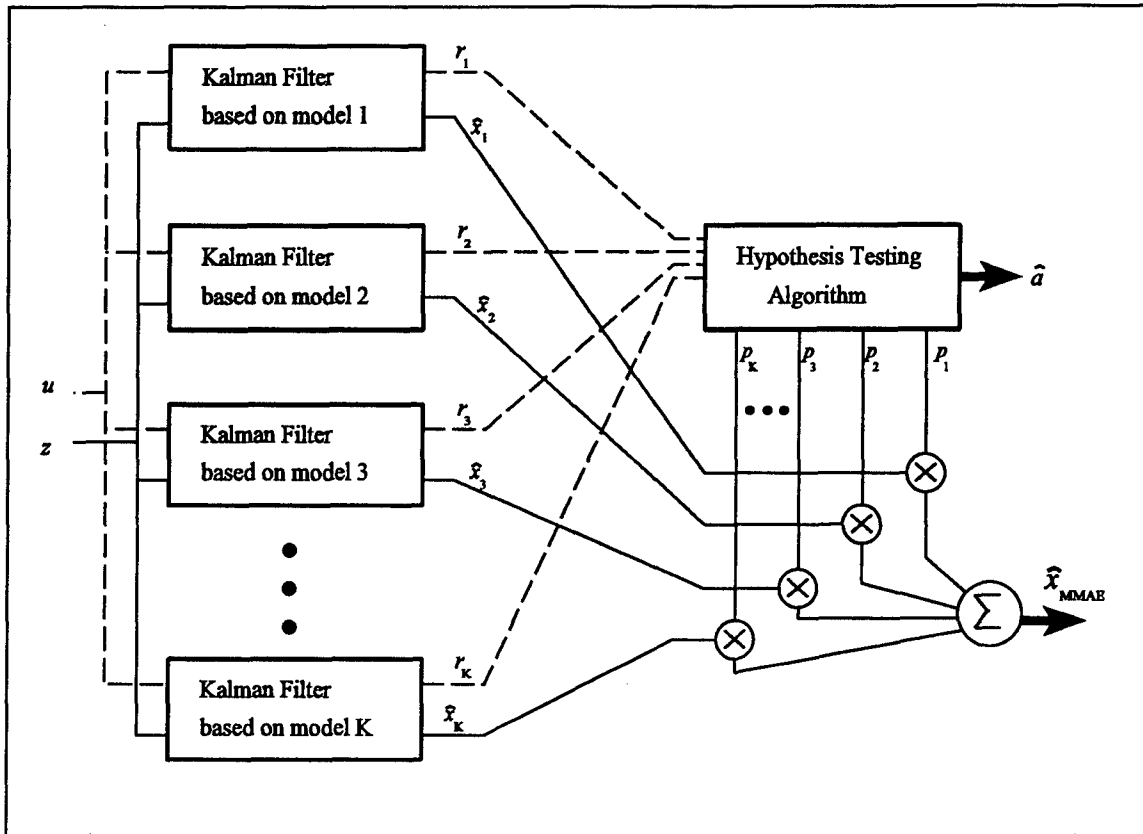


Figure 4. Multiple Model Adaptive Estimation Algorithm

The internal models of the Kalman filters can be represented by a discrete value of a parameter vector ( $a_k$ ). The Kalman filters are provided a measurement vector ( $z$ ) and the input vector ( $u$ ), and produce a state estimate ( $\hat{x}_k$ ) and a residual ( $r_k$ ). The hypothesis testing algorithm uses the residuals

to compute conditional probabilities ( $p_k$ ) of the various hypotheses that are modeled in the Kalman filters conditioned on the history of measurements received up to that time, and an estimate of the true parameter vector ( $\hat{a}$ ). For the flight control failure application in which we are interested, each Kalman filter has a different failure model ( $a_k$ ) that it uses to form the state estimate ( $\hat{x}_k$ ) and the residual ( $r_k$ ). The hypothesis testing algorithm produces an estimate of the failure status of the flight control system ( $\hat{a}$ ). The Kalman filter equations will be presented in Section 3.2. In Sections 3.2.2 through 3.2.4, we will show that the magnitude of the residual from a particular filter can be used as a relative measure of the inaccuracy (conversely, the accuracy) of the failure model used by that filter. In Section 3.3 we will study various hypothesis testing algorithms that assign conditional probabilities ( $p_k$ ) to each of the hypotheses that were used to form the Kalman filter models. These conditional probabilities indicate the relative correctness of the various filter models, and can be used to select the best estimate of the true system model, weight the individual state estimates appropriately, and form a probability-weighted average state estimate ( $\hat{x}_{\text{MMAE}}$ ). The specific workings of the algorithms in these blocks, along with various modifications to those algorithms, are examined in the following sections.

## 3.2 Kalman Filter Bank

3.2.1 Basic Equations. The Kalman filter algorithm uses a model of the true system to generate estimates of the noise-corrupted measurements before they are taken at a particular sample time, assuming that the measurements are linear combinations of the system states. Previous researchers have computed the mean and covariance of the residual under the assumption that the Kalman filter model accurately represents the true system. We will relax this assumption by allowing certain differences between the Kalman filter model and a model that accurately represents the true

system. Under these different assumptions, we will compute the mean and covariance of the Kalman filter residual.

We assume that we have noise-corrupted measurements of a possibly nonlinear system that we are examining. These true system measurements will be denoted  $z(t_i)$ . We also assume that there exists a linear model that will produce accurate representations of the true system measurements. This model will be called the true system model and denoted with the subscript  $T$ . Thus the representations of the true system measurements will be denoted  $z_T(t_i)$ , and we are assuming that these true system model representations portray the true system measurements so that  $z(t_i) = z_T(t_i)$ . Now assume that the true system model is a discrete-time equivalent system model [33] of the form

$$\begin{aligned} \mathbf{x}_T(t_i) &= \Phi_T \mathbf{x}_T(t_{i-1}) + \mathbf{B}_T \mathbf{u}(t_{i-1}) + \mathbf{G}_T \mathbf{w}_T(t_{i-1}) \\ \mathbf{z}(t_i) &= \mathbf{z}_T(t_i) = \mathbf{H}_T \mathbf{x}_T(t_i) + \mathbf{v}_T(t_i) \end{aligned} \quad (1)$$

where  $\mathbf{x}_T$  is the true system state vector

$\Phi_T$  is the state transition matrix, the discrete-time equivalent of the true system dynamics matrix

$\mathbf{B}_T$  is the discrete-time equivalent of the true system control input matrix

$\mathbf{u}$  is the system input vector

$\mathbf{G}_T$  is the discrete-time equivalent true noise input matrix

$\mathbf{w}_T$  is an additive white discrete-time dynamics noise input with zero mean and

$$E\{\mathbf{w}_T(t_i) \mathbf{w}_T^T(t_j)\} = \begin{cases} \mathbf{Q}_T, & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (2)$$

$z_T$  is the true system measurement vector

$H_T$  is the true system output matrix

$v_T$  is an additive white measurement noise input, independent of  $w_T$ , with zero mean and

$$E \{ v_T(t_i) v_T^T(t_j) \} = \begin{cases} R_T & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (3)$$

We are assuming that the sampling rate is sufficiently high so that the system modes and noise models can be considered constant over several sampling periods. Thus the  $\Phi_T, B_T, G_T, H_T, Q_T$ , and  $R_T$  matrices are assumed to be time invariant over the duration of the simulations used in this research. Also, we have assumed that the system control input is held constant over each time sample.

The assumption that the system model is time invariant allows us to consider using a steady state Kalman filter model (and eventually a steady state constant-gain Kalman filter for implementation), which will be denoted with the subscript  $k$ . Thus we have

$$\begin{aligned} x_k(t_i) &= \Phi_k x_k(t_{i-1}) + B_k u(t_{i-1}) + G_k w_k(t_{i-1}) \\ z_k(t_i) &= H_k x_k(t_i) + v_k(t_i) \end{aligned} \quad (4)$$

where  $x_k$  is the Kalman filter model state vector

$\Phi_k$  is the Kalman filter model state transition matrix

$B_k$  is the Kalman filter model control input matrix

$u$  is the system input vector

$G_k$  is the Kalman filter model noise input matrix

$w_k$  is an additive white discrete-time dynamics noise input used in the Kalman filter model,

with zero mean and

$$E\{w_k(t_i)w_k^T(t_j)\} = \begin{cases} Q_k, & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (5)$$

$z_k$  is the Kalman filter model measurement vector

$H_k$  is the Kalman filter model output matrix

$v_k$  is an additive white measurement noise input that is used in the Kalman filter model. This

noise input is assumed to be independent of  $w_k$ , and zero-mean with

$$E\{v_k(t_i)v_k^T(t_j)\} = \begin{cases} R_k & t_i = t_j \\ 0, & t_i \neq t_j \end{cases} \quad (6)$$

Note that the Kalman filter model and the truth model are both linear models, but the dimensionality of these two models may not necessarily be the same. In most cases, the Kalman filter model is a reduced order (the number of Kalman filter model states is often a subset of the truth model states) version of the truth model.

The Kalman filter algorithm uses this model to define time propagation and measurement update equations of the Kalman filter state estimates and the Kalman filter state estimate covariance matrix. The Kalman filter state estimate propagation equation based on the Kalman filter model is:

$$\begin{aligned} \hat{x}_k(t_i^-) &= \Phi_k \hat{x}_k(t_{i-1}^-) + B_k u(t_{i-1}) \\ \hat{z}_k(t_i^-) &= H_k \hat{x}_k(t_i^-) \end{aligned} \quad (7)$$

where  $\hat{x}_k$  is the Kalman filter state estimate vector

$\hat{z}_k(t_i^-)$  is the Kalman filter estimate of the measurement vector before it becomes available

$t_i^-$  is the time just before the measurement update at the  $i$ th time sample, and

$t_{i-1}^+$  is the time just after the measurement update at the  $(i - 1)$  time sample,

and the state estimate covariance matrix propagation equation:

$$P_k(t_i^-) = \Phi_k P_k(t_{i-1}^+) \Phi_k^T + G_k Q_k G_k^T. \quad (8)$$

The Kalman filter state estimates are updated using:

$$\hat{x}_k(t_i^+) = \hat{x}_k(t_i^-) + K_k(t_i) r_k(t_i) \quad (9)$$

where the Kalman filter gain is:

$$K_k(t_i) = P_k(t_i^-) H_k^T A_k(t_i)^{-1} \quad (10)$$

and the Kalman filter-computed residual covariance matrix  $A_k$  is:

$$A_k(t_i) = H_k P_k(t_i^-) H_k^T + R_k \quad (11)$$

where  $R_k$  is the measurement noise covariance matrix used in the Kalman filter model.

The Kalman filter residual vector, shown in Eq (9), is defined as:

$$r_k(t_i) \triangleq z(t_i) - H_k \hat{x}_k(t_i^-) = z_T(t_i) - H_k \hat{x}_k(t_i^-) \quad (12)$$

which is simply the difference between the measurements ( $z$ ) and the Kalman filter estimates, based on its model, of those measurements before they are taken ( $H_k \hat{x}_k(t_i^-)$ ).

The Kalman filter state estimate covariance matrix is updated using:

$$P_k(t_i^+) = P_k(t_i^-) - K_k(t_i) H_k P_k(t_i^-) . \quad (13)$$

Poor numerical characteristics of these equations (particularly the last one) require that this algorithm be implemented using the U-D (unitary upper-triangular - diagonal) covariance factorization form. The development of this factorization form can be found in Maybeck [33].

The steady state values of the Kalman filter estimate of the state covariance matrix can be precomputed by iterating Eqs (8), (10), (11), and (13) until steady state of the covariance and gain matrices is reached. Once this value for the state covariance matrix is found, the steady state Kalman filter gain  $K_k$  and the steady state Kalman filter residual covariance matrix  $A_k$  are computed using Eq (10) and Eq (11). With this steady state implementation, the state covariance matrix, the steady state Kalman filter gain, and the steady state Kalman filter residual covariance matrices are assumed to be constant and therefore do not need to be computed in real time. The steady state Kalman filter equations become:

$$\hat{x}_k(t_i^-) = \Phi_k \hat{x}_k(t_{i-1}^+) + B_k u(t_{i-1}) \quad (14)$$

for propagating the state estimates and

$$\hat{x}_k(t_i^+) = \hat{x}_k(t_i^-) + K_k r_k(t_i) \quad (15)$$

for updating the state estimates.

**3.2.2 Nomenclature for Representing Mismodeling.** We will be investigating the effects of an incorrect Kalman filter model on the filter's residual. Specifically we will look at incorrect modeling of the state transition matrix,  $\Phi$ , the output matrix,  $H$ , and the input

matrix,  $B$ . We also are assuming that the Kalman filter model and the truth model dynamics noise strength,  $Q$ , measurement noise strength,  $R$ , and noise input matrices,  $G$ , are equivalent. These conditions were chosen because they commonly occur in failure detection applications where MMAE is used.

One such application is identifying flight control failures, where a mismodeled input matrix is used to represent an actuator failure and a mismodeled output matrix would be a sensor failure. For example, a single actuator failure can be represented by zeroing a column of the input matrix,  $B$ . The result is that the element of the system input,  $u$ , that corresponds to that column of  $B$  will have no effect on the system dynamics, which is exactly what we would expect for a failed actuator. Likewise, a single sensor failure can be represented by zeroing a row of the output matrix,  $H$ . The result is that the element of the measurement that corresponds to the zeroed row of the output matrix will not consist of a linear combination of the system states, but will only have the additive white measurement noise  $v_r$ , which is precisely the expected result of a failed sensor.

For example, assume that the truth model represents the case of a single actuator failure and the Kalman filter model assumes a non-failed system; the obvious result will be a difference between the truth model and the Kalman filter model. In this case the difference occurs in the column of the system input matrix that corresponds to the failed actuator. The truth model would have a column of zeros, while the filter model would have the non-failed terms (some would be nonzero). Similarly, a failed sensor truth model would result in a difference in the row of the output matrix.

A mismodeling of the state transition matrix can occur when the filter model is based on a certain operating point (altitude, velocity, center of gravity, etc.), while the true operating point is different. The system dynamics are directly related to the operating point. Thus, there would be a

difference between the filter model and truth model, certainly in the state transition matrix and possibly in the system input matrix.

In most applications of Kalman filtering, a designer will decrease the order (the number of states) of the Kalman filter design model from that of the truth model, to reduce the computation loading required to implement the Kalman filter. This is done by eliminating certain states that do not significantly impact the fidelity of the filter model, and then appropriately adding pseudonoise to the filter model to account for the added uncertainty caused by the elimination of these states. The result of eliminating these states is an incorrectly modeled state transition matrix, output matrix, and input matrix. For example, if we were eliminating the  $n^{\text{th}}$  state in the following model:

$$\begin{aligned} \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{n-1}(t_i^-) \\ \hat{x}_n(t_i^-) \end{bmatrix} &= \begin{bmatrix} \phi_{1,1} \cdots \phi_{1,n-1} & \phi_{1,n} \\ \vdots & \vdots \\ \phi_{n-1,1} \cdots \phi_{n-1,n-1} & \phi_{n-1,n} \\ \phi_{n,1} \cdots \phi_{n,n-1} & \phi_{n,n} \end{bmatrix} \begin{bmatrix} \hat{x}_1(t_{i-1}^+) \\ \vdots \\ \hat{x}_{n-1}(t_{i-1}^+) \\ \hat{x}_n(t_{i-1}^+) \end{bmatrix} + \begin{bmatrix} b_{1,1} \cdots b_{1,r-1} & b_{1,r} \\ \vdots & \vdots \\ b_{n-1,1} \cdots b_{n-1,r-1} & b_{n-1,r} \\ b_{n,1} \cdots b_{n,r-1} & b_{n,r} \end{bmatrix} \begin{bmatrix} u_1(t_{i-1}) \\ \vdots \\ u_{r-1}(t_{i-1}) \\ u_r(t_{i-1}) \end{bmatrix} \\ \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{m-1}(t_i^-) \\ \hat{x}_m(t_i^-) \end{bmatrix} &= \begin{bmatrix} h_{1,1} \cdots h_{1,n-1} & h_{1,n} \\ \vdots & \vdots \\ h_{m-1,1} \cdots h_{m-1,n-1} & h_{m-1,n} \\ h_{m,1} \cdots h_{m,n-1} & h_{m,n} \end{bmatrix} \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{n-1}(t_i^-) \\ \hat{x}_n(t_i^-) \end{bmatrix} \end{aligned} \quad (16)$$

we would need to zero out the  $n^{\text{th}}$  column and  $n^{\text{th}}$  row of the state transition matrix, the  $n^{\text{th}}$  row of the system input matrix and the  $n^{\text{th}}$  column of the output matrix. This results in:

$$\begin{aligned} \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{n-1}(t_i^-) \\ 0 \end{bmatrix} &= \begin{bmatrix} \phi_{1,1} \cdots \phi_{1,n-1} & 0 \\ \vdots & \vdots \\ \phi_{n-1,1} \cdots \phi_{n-1,n-1} & 0 \\ 0 \cdots 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t_{i-1}^+) \\ \vdots \\ \hat{x}_{n-1}(t_{i-1}^+) \\ \hat{x}_n(t_{i-1}^+) \end{bmatrix} + \begin{bmatrix} b_{1,1} \cdots b_{1,r-1} & b_{1,r} \\ \vdots & \vdots \\ b_{n-1,1} \cdots b_{n-1,r-1} & b_{n-1,r} \\ 0 \cdots 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t_{i-1}) \\ \vdots \\ u_{r-1}(t_{i-1}) \\ u_r(t_{i-1}) \end{bmatrix} \\ \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{m-1}(t_i^-) \\ \hat{x}_m(t_i^-) \end{bmatrix} &= \begin{bmatrix} h_{1,1} \cdots h_{1,n-1} & 0 \\ \vdots & \vdots \\ h_{m-1,1} \cdots h_{m-1,n-1} & 0 \\ h_{m,1} \cdots h_{m,n-1} & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_1(t_i^-) \\ \vdots \\ \hat{x}_{n-1}(t_i^-) \\ \hat{x}_n(t_i^-) \end{bmatrix} \end{aligned} \quad (17)$$

which essentially eliminates the  $n^{\text{th}}$  state.

This method of eliminating states results in partitions of zeros in the model, which most designers would simply drop from the model so that the truth model and the Kalman filter models are of differing dimensions. Additionally, in many cases the designer will aggregate states so that there will not be a one-to-one correspondence between the states of the truth model and the Kalman filter models. An elegant method that allows this state reduction is described by Sheldon [55, 56]. He uses a transformation matrix that actually removes or combines various truth model states, so that the truth model and the Kalman filter model differ in dimensionality. To simplify the development of this research, I have chosen to keep the dimensionality of the truth model and the Kalman filter model the same. Further research is needed to develop the hypothesis testing techniques using the transformation matrix method described by Sheldon.

We introduce the following definitions:

$$\begin{aligned}\Delta \mathbf{B}_k &\triangleq \mathbf{B}_T - \mathbf{B}_k \rightarrow \mathbf{B}_k = \mathbf{B}_T - \Delta \mathbf{B}_k \\ \Delta \mathbf{H}_k &\triangleq \mathbf{H}_T - \mathbf{H}_k \rightarrow \mathbf{H}_k = \mathbf{H}_T - \Delta \mathbf{H}_k \\ \Delta \Phi_k &\triangleq \Phi_T - \Phi_k \rightarrow \Phi_k = \Phi_T - \Delta \Phi_k\end{aligned}\tag{18}$$

where we are implicitly assuming that the dimensions of the filter model and the truth model are the same.

Using this definition for the examples above, we see that in most cases rather sparse matrices result. For instance, for the actuator failure where the difference between the truth model and filter model is due to a zeroed column of the  $\mathbf{B}$  matrix, the result would be:

$$\begin{aligned}\Delta \mathbf{B}_k &\triangleq \mathbf{B}_T - \mathbf{B}_k = \begin{bmatrix} b_{1,1} & \dots & b_{1,l-1} & 0 & b_{1,l+1} & \dots & b_{1,n} \\ & \vdots & & \vdots & & \vdots & \\ b_{n,1} & \dots & b_{n,l-1} & 0 & b_{n,l+1} & \dots & b_{n,n} \end{bmatrix} - \begin{bmatrix} b_{1,1} & \dots & b_{1,l} & \dots & b_{1,n} \\ & \vdots & & \vdots & \\ b_{n,1} & \dots & b_{n,l} & \dots & b_{n,n} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & -b_{1,l} & 0 & \dots & 0 \\ & \vdots & & \vdots & & \vdots & \\ 0 & \dots & 0 & -b_{n,l} & 0 & \dots & 0 \end{bmatrix} \\ \Delta \mathbf{H}_k &\triangleq \mathbf{H}_T - \mathbf{H}_k = \mathbf{0} \\ \Delta \Phi_k &\triangleq \Phi_T - \Phi_k = \mathbf{0}.\end{aligned}\tag{19}$$

Similarly, for a sensor failure where the difference between the truth model and the filter model is due to a zeroed row of the  $H$  matrix, the result would be:

$$\begin{aligned}
\Delta B_k &\triangleq B_T - B_k = 0 \\
\Delta H_k &\triangleq H_T - H_k = \begin{bmatrix} h_{1,1} & \dots & h_{1,n} \\ h_{l-1,1} & \dots & h_{l-1,n} \\ 0 & \dots & 0 \\ h_{l+1,1} & \dots & h_{l+1,n} \\ \vdots & \dots & \vdots \\ h_{j,1} & \dots & h_{j,n} \end{bmatrix} - \begin{bmatrix} h_{1,1} & \dots & h_{1,n} \\ h_{l-1,1} & \dots & h_{l-1,n} \\ h_{l,1} & \dots & h_{l,n} \\ h_{l+1,1} & \dots & h_{l+1,n} \\ \vdots & \dots & \vdots \\ h_{j,1} & \dots & h_{j,n} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & 0 \\ -h_{l,1} & \dots & -h_{l,n} \\ 0 & \dots & 0 \\ \vdots & \dots & \vdots \end{bmatrix} \\
\Delta \Phi_k &\triangleq \Phi_T - \Phi_k = 0.
\end{aligned} \tag{20}$$

Initially, we will assume that the true system model is accurately represented in one of the Kalman filters of the MMAE. This assumption allows us to define the true system Kalman filter gain,  $K_T$ , which is the steady state Kalman filter gain of the filter that uses the true system model. The Kalman filter that uses the true system model will be called the true filter. Later, we will relax this assumption and discuss the effect on the performance of the MMAE.

Equations (5), (7), (8), and (10) show that the steady state Kalman filter gain is a function of  $\Phi_k$ ,  $G_k$ ,  $H_k$ ,  $Q_k$ , and  $R_k$ . Note that it is not a function of  $B_k$ , thus any mismodeling in the input matrix will not change the steady state Kalman filter gain. However, any difference between  $H_k$  and  $H_T$ , or  $\Phi_k$  and  $\Phi_T$ , will result in a difference in Kalman filter gains between the true filter and the filter with the mismodeling. Therefore we define

$$\Delta K_k \triangleq K_T - K_k \rightarrow K_k = K_T - \Delta K_k. \tag{21}$$

We now define the error between the Kalman filter state estimate and the true state as:

$$\epsilon_k(t_i^-) \triangleq x_T(t_i) - \hat{x}_k(t_i^-), \tag{22}$$

and

$$\epsilon_k(t_i^+) \triangleq x_T(t_i) - \hat{x}_k(t_i^+). \quad (23)$$

For the case in which the Kalman filter model and the true system model match, that is the true filter, the state estimate error is defined as:

$$\epsilon_T(t_i^-) \triangleq x_T(t_i) - \hat{x}_T(t_i^-), \quad (24)$$

and

$$\epsilon_T(t_i^+) \triangleq x_T(t_i) - \hat{x}_T(t_i^+). \quad (25)$$

Note that until the filter state estimates are updated using  $z(t_i)$ , the best estimate of  $x_T(t_i)$  is  $\hat{x}_T(t_i^-)$ . Therefore:

$$E_{Z(t_{i-1})} \{ x_T(t_i) \} = \hat{x}_T(t_i^-) \quad (26)$$

where  $E_{Z(t_{i-1})} \{ \cdot \}$  is the conditional expectation operator, which is conditioned on  $Z(t_{i-1})$ , the history of measurements up to and including time  $t_{i-1}$ .

The true filter state estimate covariance matrix is defined as:

$$\begin{aligned} P_T(t_i^-) &\triangleq E_{Z(t_{i-1})} \{ [x_T(t_i) - \hat{x}_T(t_i^-)] [x_T(t_i) - \hat{x}_T(t_i^-)]^T \} \\ &= E_{Z(t_{i-1})} \{ \epsilon_T(t_i^-) \epsilon_T(t_i^-)^T \}. \end{aligned} \quad (27)$$

### 3.2.3 Covariance of the Residual. We will derive the covariance of the residual by

first deriving an expression for the mean of the residual. Starting with Eq (9) we get:

$$\begin{aligned}
E_{Z(t_{i-1})} \{ r_k(t_i) \} &= E_{Z(t_{i-1})} \{ [ z(t_i) - H_k \hat{x}_k(t_i^-) ] \} \\
&= E_{Z(t_{i-1})} \{ z(t_i) \} - E_{Z(t_{i-1})} \{ H_k \hat{x}_k(t_i^-) \}.
\end{aligned} \tag{28}$$

Note that we are conditioning the expectation of the measurement history up to the  $t_{i-1}$  time sample. Eqs (4), (6), (7), (8), and (9) show that the state estimates in the second term of Eq (28) are simply a function of the Kalman filter model and the previous measurements. Therefore, if we are given this measurement history, the state estimates are not random and can be directly computed, so the only random variable in Eq (28) is  $z(t_i)$ . Thus we have:

$$E_{Z(t_{i-1})} \{ r_k(t_i) \} = E_{Z(t_{i-1})} \{ z(t_i) \} - H_k \hat{x}_k(t_i^-). \tag{29}$$

Now we use Eq (1) to get:

$$\begin{aligned}
E_{Z(t_{i-1})} \{ z(t_i) \} &= E_{Z(t_{i-1})} \{ H_T x_T(t_i) + v(t_i) \} \\
&= H_T E_{Z(t_{i-1})} \{ x_T(t_i) \} + E_{Z(t_{i-1})} \{ v(t_i) \} \\
&= H_T \hat{x}_T(t_i^-) + 0 \\
E_{Z(t_{i-1})} \{ z(t_i) \} &= H_T \hat{x}_T(t_i^-).
\end{aligned} \tag{30}$$

Now to compute the conditional covariance matrix of the residual, conditioned on the measurement history up to the  $t_{i-1}$  time sample:

$$\begin{aligned}
cov_{Z(t_{i-1})} \{ r_k(t_i) \} &= E_{Z(t_{i-1})} \{ r_k(t_i) r_k(t_i)^T \} - E_{Z(t_{i-1})} \{ r_k(t_i) \} E_{Z(t_{i-1})} \{ r_k(t_i)^T \} \\
&= E_{Z(t_{i-1})} \{ [ z(t_i) - H_k \hat{x}_k(t_i^-) ] [ z(t_i) - H_k \hat{x}_k(t_i^-) ]^T \} \\
&\quad - [ E_{Z(t_{i-1})} \{ z(t_i) \} - H_k \hat{x}_k(t_i^-) ] [ E_{Z(t_{i-1})} \{ z(t_i) \} - H_k \hat{x}_k(t_i^-) ]^T
\end{aligned} \tag{31}$$

$$\begin{aligned}
\text{cov}_{Z(t_{i-1})}\{r_k(t_i)\} &= E_{Z(t_{i-1})}\{z(t_i)z(t_i)^T\} - E_{Z(t_{i-1})}\{z(t_i)\} \hat{x}_k(t_i^-)^T H_k^T \\
&\quad - H_k \hat{x}_k(t_i^-) E_{Z(t_{i-1})}\{z(t_i)^T\} + H_k \hat{x}_k(t_i^-) \hat{x}_k(t_i^-)^T H_k^T \\
&\quad - \left[ E_{Z(t_{i-1})}\{z(t_i)\} E_{Z(t_{i-1})}\{z(t_i)^T\} - E_{Z(t_{i-1})}\{z(t_i)\} \hat{x}_k(t_i^-)^T H_k^T \right. \\
&\quad \left. - H_k \hat{x}_k(t_i^-) E_{Z(t_{i-1})}\{z(t_i)^T\} + H_k \hat{x}_k(t_i^-) \hat{x}_k(t_i^-)^T H_k^T \right] \\
&= E_{Z(t_{i-1})}\{z(t_i)z(t_i)^T\} - E_{Z(t_{i-1})}\{z(t_i)\} E_{Z(t_{i-1})}\{z(t_i)^T\} \\
&= \text{cov}_{Z(t_{i-1})}\{z(t_i)\}
\end{aligned} \tag{32}$$

This last expression tells us that the covariance matrix of the residual is not dependent on the Kalman filter model, it is only dependent on the covariance of the measurements. We would expect this because, as noted earlier, the only random part of the residual, given the measurement history up to  $t_{i-1}$ , is  $z(t_i)$ . Thus, in the MMAE filter bank, all of the Kalman filter residuals would have the same covariance since they all use the same measurement vector,  $z(t_i)$ , and measurement history,  $Z(t_{i-1})$ .

To get a more explicit expression for the covariance matrix of the residual, we compute the conditional covariance matrix of  $z(t_i)$  and use Eq (1) to get:

$$\begin{aligned}
\text{cov}_{Z(t_{i-1})}\{r_k(t_i)\} &= \text{cov}_{Z(t_{i-1})}\{z(t_i)\} \\
&= E_{Z(t_{i-1})}\{z(t_i)z(t_i)^T\} - E_{Z(t_{i-1})}\{z(t_i)\} E_{Z(t_{i-1})}\{z(t_i)^T\} \\
&= E_{Z(t_{i-1})}\left\{ \left[ H_T x_T(t_i) + v_T(t_i) \right] \left[ H_T x_T(t_i) + v_T(t_i) \right]^T \right\} - H_T \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T H_T^T \\
&= H_T E_{Z(t_{i-1})}\{x_T(t_i)x_T(t_i)^T\} H_T^T + H_T \hat{x}_T(t_i^-) E_{Z(t_{i-1})}\{v_T(t_i)^T\} \\
&\quad + E_{Z(t_{i-1})}\{v_T(t_i)\} \hat{x}_T(t_i^-)^T H_T^T + E_{Z(t_{i-1})}\{v_T(t_i)v_T(t_i)^T\} \\
&\quad - H_T \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T H_T^T \\
&= H_T E_{Z(t_{i-1})}\{x_T(t_i)x_T(t_i)^T\} H_T^T + R_T(t_i) - H_T \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T H_T^T
\end{aligned} \tag{33}$$

$$\begin{aligned}
\text{cov}_{Z(t_{i-1})}\{r_k(t_i)\} &= H_T \left[ E_{Z(t_{i-1})}\{x_T(t_i) x_T(t_i)^T\} - \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T \right. \\
&\quad \left. - \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T + \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T \right] H_T^T + R_T(t_i) \\
&= H_T \left[ E_{Z(t_{i-1})}\{x_T(t_i) x_T(t_i)^T\} - \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T \right. \\
&\quad \left. - \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T + \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T \right] H_T^T + R_T(t_i) \\
&= H_T \left[ E_{Z(t_{i-1})}\{x_T(t_i) x_T(t_i)^T\} - E_{Z(t_{i-1})}\{x_T(t_i)\} \hat{x}_T(t_i^-)^T \right. \\
&\quad \left. - \hat{x}_T(t_i^-) E_{Z(t_{i-1})}\{x_T(t_i)^T\} + \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T \right] H_T^T + R_T(t_i) \\
&= H_T E_{Z(t_{i-1})}\{x_T(t_i) x_T(t_i)^T - x_T(t_i) \hat{x}_T(t_i^-)^T - \hat{x}_T(t_i^-) x_T(t_i)^T + \hat{x}_T(t_i^-) \hat{x}_T(t_i^-)^T\} H_T^T \\
&\quad + R_T(t_i) \\
&= H_T E_{Z(t_{i-1})}\{[x_T(t_i) - \hat{x}_T(t_i^-)][x_T(t_i) - \hat{x}_T(t_i^-)]^T\} H_T^T + R_T(t_i) \\
&= H_T P_T(t_i^-) H_T^T + R_T(t_i)
\end{aligned}$$

$$\text{cov}_{Z(t_{i-1})}\{r_k(t_i)\} = A_T(t_i) \tag{34}$$

where  $A_T(t_i)$  denotes the Kalman filter residual covariance matrix at time  $t_i$ . We now assume that the Kalman filter has reached steady state so that the covariance matrix is constant. We denote the covariance matrix of the residual of the steady state Kalman filter that is using the true system model as  $A_T$ . This result tells us that the steady state covariance matrix of any Kalman filter residual is independent of the Kalman filter model and can be precomputed simply by calculating the steady state residual covariance of a Kalman filter with the true model.

**3.2.4 Mean of the Residual.** Earlier it was shown that the conditional covariance of the residual from any of the Kalman filters in the MMAE is dependent only on the conditional covariance of the actual measurements; thus any mismodeling in the design model upon which the

filter is based, has no effect on the conditional covariance of the residual. We will now examine the effects of mismodeling on the conditional mean of the residual.

First, we expand one of the terms in Eq (12) using the definition in Eq (1) to get:

$$\begin{aligned}
r_k(t_i) &\triangleq z(t_i) - H_k \hat{x}_k(t_i) = [H_T x_T(t_i) + v_T(t_i)] - H_k \hat{x}_k(t_i) \\
&= H_T [\Phi_T x_T(t_{i-1}) + B_T u(t_{i-1}) + G_T w_d(t_{i-1})] + v_T(t_i) - H_k [\Phi_k \hat{x}_k(t_{i-1}) + B_k u(t_{i-1})] \\
&= H_T \Phi_T x_T(t_{i-1}) - H_k \Phi_k \hat{x}_k(t_{i-1}) + [H_T B_T - H_k B_k] u(t_{i-1}) + H_T G_T w_d(t_{i-1}) + v_T(t_i) \\
&= H_T \Phi_T x_T(t_{i-1}) - (H_T - \Delta H_k) (\Phi_T - \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + [H_T B_T - (H_T - \Delta H_k) (B_T - \Delta B_k)] u(t_{i-1}) + H_T G_T w_d(t_{i-1}) + v_T(t_i) \\
&= H_T \Phi_T [x_T(t_{i-1}) - \hat{x}_k(t_{i-1})] + (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta \Phi_k) u(t_{i-1}) + H_T G_T w_d(t_{i-1}) + v_T(t_i) \\
&= H_T \Phi_T \epsilon_k(t_{i-1}) + (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta \Phi_k) u(t_{i-1}) + H_T G_T w_d(t_{i-1}) + v_T(t_i)
\end{aligned} \tag{35}$$

We take the expectation of this to find the mean of the residual:

$$\begin{aligned}
E_{Z(t_{i-1})} \{r_k(t_i)\} &= E_{Z(t_{i-1})} \{H_T \Phi_T \epsilon_k(t_{i-1}) + (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta \Phi_k) u(t_{i-1}) + H_T G_T w_d(t_{i-1}) + v_T(t_i)\} \\
&= H_T \Phi_T E_{Z(t_{i-1})} \{\epsilon_k(t_{i-1})\} + (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta \Phi_k) u(t_{i-1}) + H_T G_T E_{Z(t_{i-1})} \{w_d(t_{i-1})\} \\
&\quad + E_{Z(t_{i-1})} \{v_T(t_i)\} \\
E_{Z(t_{i-1})} \{r_k(t_i)\} &= H_T \Phi_T E_{Z(t_{i-1})} \{\epsilon_k(t_{i-1})\} \\
&\quad + (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}) \\
&\quad + (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta \Phi_k) u(t_{i-1})
\end{aligned} \tag{36}$$

To derive an expression for the first term, we take the conditional expectation of Eq (23) and then step back one data sample in time:

$$\begin{aligned}
E_{Z(t_{i-1})}\{\epsilon_k(t_i^*)\} &= E_{Z(t_{i-1})}\{x_T(t_i) - \hat{x}_k(t_i^*)\} \\
&= E_{Z(t_{i-1})}\{x_T(t_i) - [\hat{x}_k(t_i^*) + K_k r_k(t_i)]\} \\
&= E_{Z(t_{i-1})}\{x_T(t_i) - [\hat{x}_k(t_i^*) - K_k H_k \hat{x}_k(t_i^*) + K_k z(t_i)]\} \\
&= E_{Z(t_{i-1})}\{x_T(t_i) - (I - K_k H_k) \hat{x}_k(t_i^*) - K_k [H_T x_T(t_i) + v_T(t_i)]\} \\
&= E_{Z(t_{i-1})}\{(I - K_k H_T) x_T(t_i) - (I - K_k H_k) \hat{x}_k(t_i^*)\} - K_k E_{Z(t_{i-1})}\{v_T(t_i)\} \\
&= E_{Z(t_{i-1})}\{(I - K_k H_T) [\Phi_T x_T(t_{i-1}) + B_T u(t_{i-1}) + G_T w_d(t_{i-1})] \\
&\quad - (I - K_k H_k) [\Phi_k \hat{x}_k(t_{i-1}) + B_k u(t_{i-1})])\} \\
&= E_{Z(t_{i-1})}\{(I - K_k H_T) \Phi_T x_T(t_{i-1}) - (I - K_k H_k) \Phi_k \hat{x}_k(t_{i-1})\} \\
&\quad + [(I - K_k H_T) B_T - (I - K_k H_k) B_k] u(t_{i-1}) \\
&\quad + (I - K_k H_T) G_T E_{Z(t_{i-1})}\{w_d(t_{i-1})\} \\
&= E_{Z(t_{i-1})}\{(I - K_k H_T) \Phi_T x_T(t_{i-1}) - (I - K_k H_k) \Phi_k \hat{x}_k(t_{i-1})\} \\
&\quad + [(I - K_k H_T) B_T - (I - K_k H_k) B_k] u(t_{i-1}) \tag{37}
\end{aligned}$$

Now we substitute in the definitions from Eq (18):

$$\begin{aligned}
E_{Z(t_{i-1})}\{\epsilon_k(t_i^*)\} &= E_{Z(t_{i-1})}\{(I - K_k H_T) \Phi_T x_T(t_{i-1}) - [I - K_k (H_T - \Delta H_k)] (\Phi_T - \Delta \Phi_k) \hat{x}_k(t_{i-1})\} \\
&\quad + \{(I - K_k H_T) B_T - [I - K_k (H_T - \Delta H_k)] (B_T - \Delta B_k)\} u(t_{i-1}) \\
&= E_{Z(t_{i-1})}\{(I - K_k H_T) \Phi_T x_T(t_{i-1}) - [(I - K_k H_T) + K_k \Delta H_k] (\Phi_T - \Delta \Phi_k) \hat{x}_k(t_{i-1})\} \\
&\quad + \{(I - K_k H_T) B_T - [(I - K_k H_T) + K_k \Delta H_k] (B_T - \Delta B_k)\} u(t_{i-1}) \\
&= (I - K_k H_T) \Phi_T E_{Z(t_{i-1})}\{x_T(t_{i-1}) - \hat{x}_k(t_{i-1})\} \\
&\quad + [(I - K_k H_T) \Delta \Phi_k - K_k \Delta H_k \Phi_k] \hat{x}_k(t_{i-1}) \\
&\quad + [(I - K_k H_T) B_T - (I - K_k H_T) B_T + (I - K_k H_T) \Delta B_k - K_k \Delta H_k B_k] u(t_{i-1})
\end{aligned}$$

$$\begin{aligned}
E_{Z(t_{i-1})} \{ \epsilon_k(t_i^*) \} &= (I - K_k H_T) \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}^*) \} \\
&+ [(I - K_k H_T) \Delta \Phi_k - K_k \Delta H_k \Phi_k] \hat{x}_k(t_{i-1}^*) \\
&+ [(I - K_k H_T) \Delta B_k - K_k \Delta H_k B_k] u(t_{i-1})
\end{aligned} \tag{38}$$

We will assume that until a certain point in time the Kalman filter model matched the true system model. For failure identification applications, this point in time would be the failure time,  $t_f$ . Up until that time, all of the modeling errors are zero and

$$\left. \begin{aligned} \hat{x}_T(t_i^*) &= \hat{x}_k(t_i^*) \\ \hat{x}_T(t_i^-) &= \hat{x}_k(t_i^-) \end{aligned} \right\} \quad \forall t_i < t_f \tag{39}$$

Therefore the mean of the state estimate error will also be zero. This assumption gives us:

$$E_{Z(t_{i-1})} \{ \epsilon_k(t_i^*) \} = \begin{cases} 0 & , \text{ for } t_i \leq t_f \\ \begin{aligned} &[(I - K_k H_T) \Delta \Phi_k - K_k \Delta H_k \Phi_k] \hat{x}_k(t_f^*) \\ &+ [(I - K_k H_T) \Delta B_k - K_k \Delta H_k B_k] u(t_f) \end{aligned} & , \text{ for } t_i = t_{f+1} \\ \begin{aligned} &(I - K_k H_T) \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}^*) \} \\ &+ [(I - K_k H_T) \Delta \Phi_k - K_k \Delta H_k \Phi_k] \hat{x}_k(t_{i-1}^*) \\ &+ [(I - K_k H_T) \Delta B_k - K_k \Delta H_k B_k] u(t_{i-1}) \end{aligned} & , \text{ for } t_i > t_{f+1} \end{cases} \tag{40}$$

Thus, to compute the mean of the residual, we keep a running calculation of the mean state estimate error using Eq (40), and then compute the residual mean using:

$$\begin{aligned}
E_{Z(t_{i-1})} \{ r_k(t_i) \} &= H_T \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}^*) \} \\
&+ (H_T \Delta \Phi_k + \Delta H_k \Phi_T - \Delta H_k \Delta \Phi_k) \hat{x}_k(t_{i-1}^*) \\
&+ (H_T \Delta B_k + \Delta H_k B_T - \Delta H_k \Delta B_k) u(t_{i-1})
\end{aligned} \tag{41}$$

We can now use this equation to look at the mean of the Kalman filter residual for several special cases of mismodeling.

#### 3.2.4.1 No Mismodeling. We will first look at the case where the Kalman

filter model and the true system model match. For this case we get:

$$\left. \begin{array}{l} B_T = B_k \\ H_T = H_k \\ \Phi_T = \Phi_k \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta B_k = 0 \\ \Delta H_k = 0 \\ \Delta \Phi_k = 0. \end{array} \right. \quad (42)$$

Using Eq (40) and Eq (41) we find that:

$$\begin{aligned} E_{Z(t_{i-1})} \{ \epsilon_k(t_i) \} &= 0 \quad \forall t \\ \rightarrow E_{Z(t_{i-1})} \{ r_k(t_i) \} &= 0 \quad \forall t \end{aligned} \quad (43)$$

which Maybeck [33] has also shown using a different development.

#### 3.2.4.2 Mismodeled Input Matrix. A mismodeled input matrix, which is

used to model an actuator failure in flight control failure identification applications, would result in:

$$\left. \begin{array}{l} B_T \neq B_k \\ H_T = H_k \\ \Phi_T = \Phi_k \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta B_k \neq 0 \\ \Delta H_k = 0 \\ \Delta \Phi_k = 0. \end{array} \right. \quad (44)$$

Using Eq (40) we get:

$$E_{Z(t_{i-1})} \{ \epsilon_k(t_i) \} = \left\{ \begin{array}{ll} 0 & , \text{ for } t_i \leq t_f \\ (I - K_k H_T) \Delta B_k u(t_f) & , \text{ for } t_i = t_{f,1} \\ (I - K_k H_T) \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \\ + (I - K_k H_T) \Delta B_k u(t_{i-1}) & , \text{ for } t_i > t_{f,1} \end{array} \right. \quad (45)$$

and from Eq (41) we get:

$$E_{Z(t_{i-1})} \{ r_k(t_i) \} = H_T \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} + H_T \Delta B_k u(t_{i-1}). \quad (46)$$

### 3.2.4.3 Mismodeled Output Matrix. A mismodeled output matrix, which

is used to model a sensor failure in flight control applications, would result in:

$$\left. \begin{array}{l} B_T = B_k \\ H_T \neq H_k \\ \Phi_T = \Phi_k \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta B_k = 0 \\ \Delta H_k \neq 0 \\ \Delta \Phi_k = 0. \end{array} \right. \quad (47)$$

Using Eq (40) again, we get:

$$E_{Z(t_{i-1})} \{ \epsilon_k(t_i) \} = \left\{ \begin{array}{ll} 0 & , \text{ for } t_i \leq t_f \\ -K_k \Delta H_k \Phi_k \hat{x}_k(t_f) - K_k \Delta H_k B_k u(t_f) & , \text{ for } t_i = t_{f,1} \\ (I - K_k H_T) \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \\ - K_k \Delta H_k \Phi_k \hat{x}_k(t_{i-1}) - K_k \Delta H_k B_k u(t_{i-1}) & , \text{ for } t_i > t_{f,1} \end{array} \right. \quad (48)$$

which can be rewritten as:

$$E_{Z(t_{i-1})} \{ \epsilon_k(t_i) \} = \left\{ \begin{array}{ll} 0 & , \text{ for } t_i \leq t_f \\ -K_k \Delta H_k \hat{x}_k(t_{f,1}) & , \text{ for } t_i = t_{f,1} \\ (I - K_k H_T) \Phi_T E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \\ - K_k \Delta H_k \hat{x}_k(t_i) & , \text{ for } t_i > t_{f,1} \end{array} \right. \quad (49)$$

Using Eq (41) we get:

$$\begin{aligned}
E_{Z(t_{i-1})}\{r_k(t_i)\} &= H_T \Phi_T E_{Z(t_{i-1})}\{\epsilon_k(t_{i-1}^*)\} + \Delta H_k \Phi_T \hat{x}_k(t_{i-1}^*) + \Delta H_k B_T u(t_{i-1}) \\
&= H_T \Phi_T E_{Z(t_{i-1})}\{\epsilon_k(t_{i-1}^*)\} + \Delta H_k \hat{x}_k(t_i^-).
\end{aligned} \tag{50}$$

#### 3.2.4.4 Mismodeled State Transition Matrix. A mismodeled state

transition matrix, which can be used to model an imperfect modeling of the aircraft modes in flight control failure identification applications, would result in:

$$\left. \begin{aligned} B_T &= B_k \\ H_T &= H_k \\ \Phi_T &\neq \Phi_k \end{aligned} \right\} \rightarrow \left\{ \begin{aligned} \Delta B_k &= 0 \\ \Delta H_k &= 0 \\ \Delta \Phi_k &\neq 0. \end{aligned} \right. \tag{51}$$

Using Eq (40) again, we get:

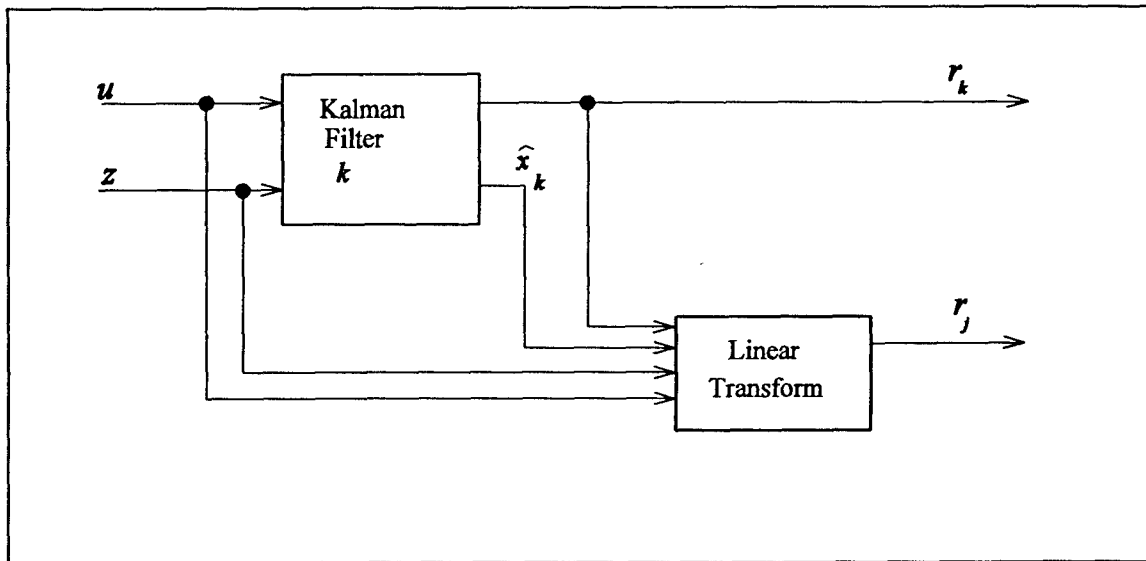
$$E_{Z(t_{i-1})}\{\epsilon_k(t_i^*)\} = \begin{cases} 0 & , \text{ for } t_i \leq t_f \\ (I - K_k H_T) \Delta \Phi_k \hat{x}_k(t_f^*) & , \text{ for } t_i = t_{f,1} \\ (I - K_k H_T) \Phi_T E_{Z(t_{i-1})}\{\epsilon_k(t_{i-1}^*)\} \\ + (I - K_k H_T) \Delta \Phi_k \hat{x}_k(t_{i-1}^-) & , \text{ for } t_i > t_{f,1} \end{cases} \tag{52}$$

Using Eq (41) we get:

$$E_{Z(t_{i-1})}\{r_k(t_i)\} = H_T \Phi_T E_{Z(t_{i-1})}\{\epsilon_k(t_{i-1}^*)\} + H_T \Delta \Phi_k \hat{x}_k(t_{i-1}^-). \tag{53}$$

#### 3.2.5 Kalman Filter Bank Implemented Using a Single Residual. In this

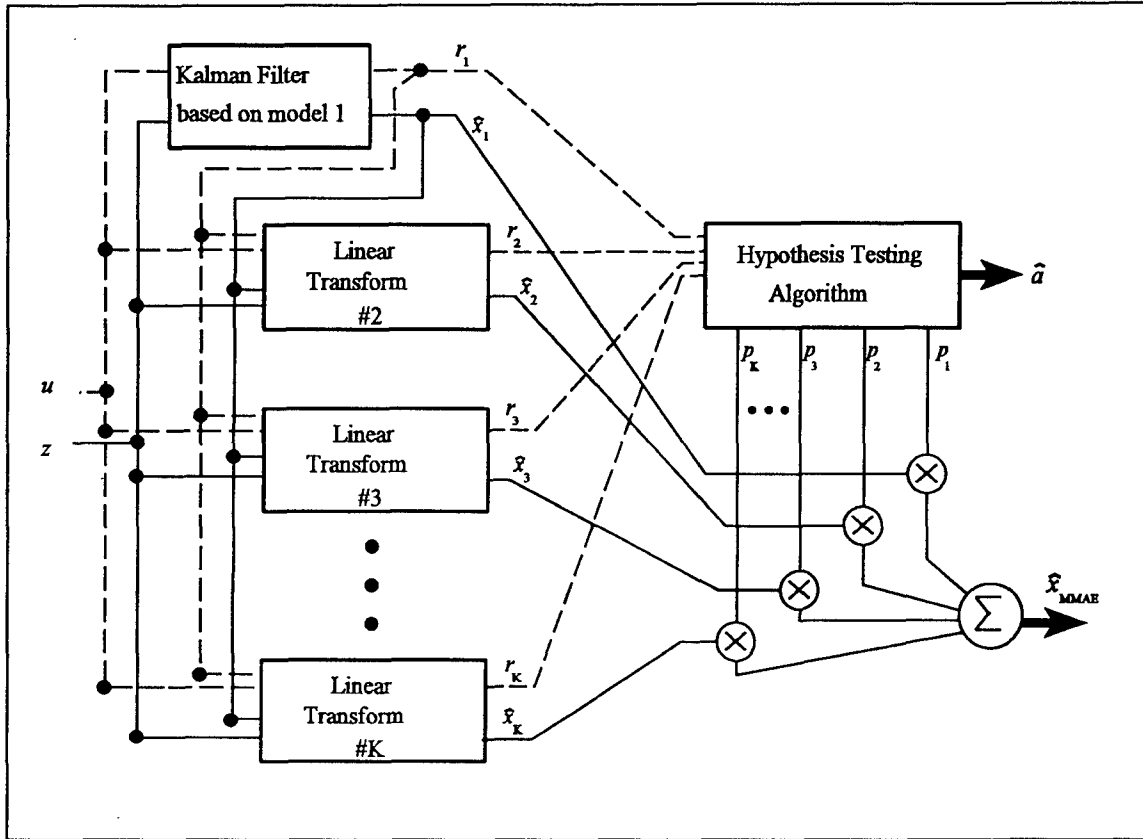
section we propose a linear transform that operates on the residual from one Kalman filter and produces the equivalent residual of another Kalman filter. The proposed structure is shown in Figure



**Figure 5 - Alternative Method of Computing a Second Kalman Filter Residual**

5. A single Kalman filter, denoted filter  $k$ , is implemented and its residual and state estimates, along with the control input and measurement vectors, are used by a linear transform to produce the equivalent of the residual from another Kalman filter, denoted filter  $j$ . This transform only requires knowing the difference between the filter models and does not require the implementation of filter  $j$ . This structure can be extended to form the equivalent of the Kalman filter bank shown in Figure 4. This alternative structure is shown in Figure 6, where a single residual is used to form the equivalent residual from all the other Kalman filters.

This structure could produce significant computational savings for many MMAE applications. We illustrate this, we will compare the operations count (number of multiplies and additions) that are need to implement the usual Kalman filter bank structure with the count needed to implement this structure. We will not assume a specific structure (i.e., diagonal, block diagonal, partitioned, etc.) of the matrices used to implement these structures, except for the  $\Delta B$ ,  $\Delta H$ , and  $\Delta \Phi$  matrices. Previously, we showed that these matrices are sparse, usually with only one nonzero row or



**Figure 6.** Alternative Multiple Model Adaptive Estimation Filter Bank Structure Using Equivalent Residuals

column. We will develop the operations count for the fully implemented Kalman filter bank in this section, and compare this to the operations count for the case of a mismodeled system input matrix (simulating an actuator failure) in Section 3.2.5.3 and a mismodeled output matrix (simulating a sensor failure) in Section 3.2.5.4.

We start by defining the various matrix dimensions as follows:  $n_s$  - number of states,  $n_i$  - number of system inputs, and  $n_m$  - number of measurements. The steady-state Kalman filter state estimates are updated using Eq (15), where the dimension of  $K_k$  is  $n_s$  by  $n_m$ , the state estimate vector is  $n_s$ , and residual vector is  $n_m$ . Thus, we will have  $n_s n_m$  multiplies, and  $n_s (n_m - 1) + n_s = n_s n_m$

additions. The state estimates are propagated using Eq (14), where the dimension of  $\Phi_k$  is  $n_s$  by  $n_s$ ,  $B_k$  is  $n_s$  by  $n_i$ , and the system input vector is  $n_i$ . Eq (14) requires  $n_s n_i + n_s n_s$  multiplies, and  $n_s (n_i - 1) + n_s (n_s - 1) + n_s = n_s n_s$  additions. Lastly, Eq (12) is used to form the current time sample of the residual, which requires  $n_s n_m$  multiplies, and  $n_m (n_s - 1) + n_m = n_s n_m$  additions. Thus, the total operations count is  $n_s (n_s + n_i + 2n_m)$  multiplies and  $n_s (n_s + n_i + 2n_m - 1)$ . The dimension for the specific application that was implemented for this research were  $n_s = 19$ ,  $n_i = 6$ , and  $n_m = 9$ , which gives us an operations count of 817 multiplies and 798 additions, per elemental filter. This operations count will be compared to the count using this alternate structure in the following sections. Now we will compute the linear transform.

3.2.5.1 Model Difference Nomenclature. This development will be similar to Section 3.2.2, except that we will be developing the difference between two Kalman filter models instead of between the true system model and a Kalman filter model. We will use the subscripts  $j$  and  $k$  to denote the two different Kalman filter models.

We are examining model differences in the state transition matrix,  $\Phi$ , the output matrix,  $H$ , and the input matrix,  $B$ . Therefore, we introduce the following definitions:

$$\begin{aligned}
 \Delta B_{kj} &\triangleq B_k - B_j \rightarrow B_j = B_k - \Delta B_{kj} \\
 \Delta H_{kj} &\triangleq H_k - H_j \rightarrow H_j = H_k - \Delta H_{kj} \\
 \Delta \Phi_{kj} &\triangleq \Phi_k - \Phi_j \rightarrow \Phi_j = \Phi_k - \Delta \Phi_{kj} \\
 \Delta K_{kj} &\triangleq K_k - K_j \rightarrow K_j = K_k - \Delta K_{kj}
 \end{aligned} \tag{54}$$

In Section 3.2.2 we defined the error between the Kalman filter state estimates and the true state using the nomenclature  $\epsilon_k$ . In Section 3.2.4 we showed that the mean error between the Kalman filter state estimates is important in computing the mean of the residual and must be computed at

each time sample to compute the mean of the residual. In a similar manner, we define the difference in state estimate errors, which is algebraically equivalent to the difference in state estimates as:

$$\begin{aligned}
\Delta \epsilon_{jk}(t_i') &\triangleq \epsilon_j(t_i') - \epsilon_k(t_i') \\
&= [x_T(t_i) - \hat{x}_j(t_i')] - [x_T(t_i) - \hat{x}_k(t_i')] \\
&= \hat{x}_k(t_i') - \hat{x}_j(t_i') \\
\rightarrow \hat{x}_j(t_i') &= \hat{x}_k(t_i') - \Delta \epsilon_{jk}(t_i').
\end{aligned} \tag{55}$$

### 3.2.5.2 Computation of the Residuals.

Our goal is to replace the  $j$ -th Kalman filter, in the MMAE filter bank, using the residual from the  $k$ -th Kalman filter. We assume that the Kalman filter states from filter  $k$  and the residual from filter  $k$  are computed at each time sample and are available for computations. Also the control input and the measurement vectors are available for computations. We assume that the model differences are only between the input matrices, output matrices, and state transition matrices (or some subset thereof). Thus, we desire to derive an expression for the residual of filter  $j$ , in terms of the state estimates of filter  $k$ , the difference between the Kalman filter state estimates of filter  $k$  and  $j$ , and the known model matrices.

We start with the definition of the residual for filter  $j$ :

$$\begin{aligned}
r_j(t_i) &\triangleq z(t_i) - H_j \hat{x}_j(t_i') \\
&= z(t_i) - H_j [\Phi_j \hat{x}_j(t_{i-1}') + B_j u(t_{i-1})] \\
&= z(t_i) - H_j \Phi_j \hat{x}_j(t_{i-1}') - H_j B_j u(t_{i-1}) \\
&= z(t_i) - H_j \Phi_j [\hat{x}_k(t_{i-1}') - \Delta \epsilon_{jk}(t_{i-1}')] - H_j B_j u(t_{i-1}) \\
r_j(t_i) &= z(t_i) - H_j \Phi_j \hat{x}_k(t_{i-1}') - H_j B_j u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}').
\end{aligned} \tag{56}$$

This expression can be used to compute the equivalent residual from filter  $j$  using the state estimates from filter  $k$ , the difference between the state estimates of filter  $k$  and filter  $j$ , and the

control input and measurement vectors. We will now derive an equivalent expression that uses the residual from filter  $k$ .

$$\begin{aligned}
r_j(t_i) &= z(t_i) - H_j \Phi_j \hat{x}_k(t_{i-1}') - H_j B_j u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
&= z(t_i) - (H_k - \Delta H_{kj})(\Phi_k - \Delta \Phi_{kj}) \hat{x}_k(t_{i-1}') \\
&\quad - (H_k - \Delta H_{kj})(B_k - \Delta B_{kj}) u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
&= z(t_i) - [H_k \Phi_k - H_j \Delta \Phi_{kj} - \Delta H_{kj} \Phi_k] \hat{x}_k(t_{i-1}') \\
&\quad - [H_k B_k - H_j \Delta B_{kj} - \Delta H_{kj} B_k] u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
&= z(t_i) - H_k [\Phi_k \hat{x}_k(t_{i-1}') + B_k u(t_{i-1})] + [H_j \Delta \Phi_{kj} + \Delta H_{kj} \Phi_k] \hat{x}_k(t_{i-1}') \\
&\quad + [H_j \Delta B_{kj} + \Delta H_{kj} B_k] u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
&= z(t_i) - H_k \hat{x}_k(t_i^-) + [H_j \Delta \Phi_{kj} + \Delta H_{kj} \Phi_k] \hat{x}_k(t_{i-1}') \\
&\quad + [H_j \Delta B_{kj} + \Delta H_{kj} B_k] u(t_{i-1}) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}'). \tag{57}
\end{aligned}$$

Thus, we get:

$$\begin{aligned}
r_j(t_i) &= r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
&\quad + [H_j \Delta \Phi_{kj} + \Delta H_{kj} \Phi_k] \hat{x}_k(t_{i-1}') + [H_j \Delta B_{kj} + \Delta H_{kj} B_k] u(t_{i-1}). \tag{58}
\end{aligned}$$

This may seem to be a more complex expression than Eq (56), but later we will see that in most cases this simplifies dramatically.

To compute the difference in state estimates between filter  $k$  and filter  $j$ , we can compute each of these estimates and then difference them. However, by doing so we would need to implement filter  $j$ , which would defeat the purpose of this derivation since we would have the residual directly from that filter. When the MMAE is initialized, the difference between the Kalman filter state estimates is either zero or known. Therefore, we need to derive a recursive expression for the

difference in state estimates between filter  $k$  and filter  $j$  ( $\Delta \epsilon_{jk}(t_i')$ ). We could then compute

$\Delta \epsilon_{jk}(t_i')$  at each time sample and then the desired residual for the filter  $j$ .

We start with Eq (55) and simply decrement back one time sample:

$$\begin{aligned}
 \Delta \epsilon_{jk}(t_i') &= \hat{x}_k(t_i') - \hat{x}_j(t_i') \\
 &= \left[ \hat{x}_k(t_i^-) + K_k r_k(t_i) \right] - \left[ \hat{x}_j(t_i^-) + K_j r_j(t_i) \right] \\
 &= \left\{ \hat{x}_k(t_i^-) + K_k \left[ z(t_i) - H_k \hat{x}_k(t_i^-) \right] \right\} - \left\{ \hat{x}_j(t_i^-) + K_j \left[ z(t_i) - H_j \hat{x}_j(t_i^-) \right] \right\} \\
 &= (I - K_k H_k) \hat{x}_k(t_i^-) - (I - K_j H_j) \hat{x}_j(t_i^-) + (K_k - K_j) z(t_i) \\
 &= (I - K_k H_k) \left[ \Phi_k \hat{x}_k(t_{i-1}') + B_k u(t_{i-1}) \right] \\
 &\quad - (I - K_j H_j) \left[ \Phi_j \hat{x}_j(t_{i-1}') + B_j u(t_{i-1}) \right] + (K_k - K_j) z(t_i) \\
 &= (I - K_k H_k) \Phi_k \hat{x}_k(t_{i-1}') - (I - K_j H_j) \Phi_j \hat{x}_j(t_{i-1}') \\
 &\quad + \left[ (I - K_k H_k) B_k - (I - K_j H_j) B_j \right] u(t_{i-1}) + (K_k - K_j) z(t_i) \tag{59}
 \end{aligned}$$

Now we substitute in the definitions from Eq (54) and Eq (55):

$$\begin{aligned}
 \Delta \epsilon_{jk}(t_i') &= (I - K_k H_k) \Phi_k \hat{x}_k(t_{i-1}') - (I - K_j H_j) \Phi_j \left[ \hat{x}_k(t_{i-1}') - \Delta \epsilon_{jk}(t_{i-1}') \right] \\
 &\quad + \left[ (I - K_k H_k) B_k - (I - K_j H_j) B_j \right] u(t_{i-1}) + \Delta K_{kj} z(t_i) \\
 &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\
 &\quad + \left[ (I - K_k H_k) \Phi_k - (I - K_j H_j) \Phi_j \right] \hat{x}_k(t_{i-1}') \\
 &\quad + \left[ (I - K_k H_k) B_k - (I - K_j H_j) B_j \right] u(t_{i-1}) + \Delta K_{kj} z(t_i) \tag{60}
 \end{aligned}$$

which is the desired recursive relationship.

Eq (58) and Eq (60) certainly appear to be very complex expressions, but under most model differences these expressions will simplify dramatically. We now look at some specific model differences to demonstrate this simplification.

### 3.2.5.3 Different Input Matrix Models. We now assume that the two

Kalman filter models differ only in the input matrix. This would represent different actuator failure conditions for the filter control failure identification application. This gives us:

$$\left. \begin{array}{l} \Phi_k = \Phi_j \\ H_k = H_j \\ K_k = K_j \\ B_k \neq B_j \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta \Phi_{kj} = 0 \\ \Delta H_{kj} = 0 \\ \Delta K_{kj} = 0 \\ \Delta B_{kj} \neq 0 \end{array} \right. \quad (61)$$

Under these assumptions Eq (60) becomes:

$$\begin{aligned} \Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\ &\quad + [(I - K_j H_j) \Phi_j - (I - K_j H_j) \Phi_j] \hat{x}_k(t_{i-1}') \\ &\quad + [(I - K_j H_j) B_k - (I - K_j H_j) B_j] u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ \Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\ &\quad + (I - K_j H_j) \Delta B_{kj} u(t_{i-1}) \end{aligned} \quad (62)$$

and Eq (58) becomes:

$$r_j(t_i) = r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + H_j \Delta B_{kj} u(t_{i-1}). \quad (63)$$

Returning to our flight control failure detection example, we find that we could construct the equivalent residuals of a bank of Kalman filters, each with a different actuator failure model, by using a single Kalman filter residual and Eqs (62) and (63). We will now compute the operations count and compare it to the required number of operations for the fully implemented Kalman filter

bank. We define  $n_B$  as the number of columns of  $\Delta B$  that are not zero, thus a mismodeling in a single column of  $B$  will produce a  $\Delta B$  with only one nonzero column, and  $n_B = 1$ . The  $(I - K_J H_J) \Phi_J$ , which is  $n_s$  by  $n_s$ , and  $(I - K_J H_J) \Delta B_{k_j}$  terms in Eq (62) are precomputable, and therefore will not be computed at each time sample. The  $(I - K_J H_J) \Delta B_{k_j}$  term is a sparse  $n_s$  by  $n_i$  matrix with only  $n_B$  nonzero columns. To evaluate the first term of Eq (62) will require  $n_s n_s$  multiplications and  $n_s(n_s-1)$  additions. The second term requires  $n_s n_B$  multiplications and  $n_s(n_B-1)$  additions. A similar analysis of Eq (63) shows that, the second term requires  $n_m n_s$  multiplications and  $n_m(n_s-1)$  additions, the third term requires  $n_m n_B$  multiplications and  $n_m(n_B-1)$  additions, plus  $2n_m$  additions to sum the first, second, and third terms together. Thus the total operations count is  $n_s (n_m + n_s + n_B) + n_m n_B$  multiplications, and  $n_m (n_B + n_s) + n_s^2 + n_s (n_B-1)$  additions. This results in 560 multiplies and 541 additions for the specific application that we implemented for this research. Subtracting this count from the total operations count, developed earlier, gives us the total operations savings of implementing this structure over the fully implemented Kalman filter structure. This savings is  $n_s (n_i + n_m) - n_B (n_s + n_m)$  multiplies and the same for the number additions, which is 257 for this particular application. This savings would be multiplied by the number of filters that are modeling actuator failures in the Kalman filter bank, to produce the total savings of implementing this structure over fully implementing the Kalman filter bank. For this specific implementation, the savings is about 30% of the required operations.

#### 3.2.5.4 Different Output Matrix Models. We now assume that the two

Kalman filter models differ only in the output matrix. This will cause the two filters to also have different Kalman filter gains, as explained in Section 3.2.2. This gives us:

$$\left. \begin{array}{l} \Phi_k = \Phi_j \\ B_k = B_j \\ H_k \neq H_j \rightarrow K_k \neq K_j \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta \Phi_{kj} = 0 \\ \Delta B_{kj} = 0 \\ \Delta H_{kj} \neq 0 \\ \Delta K_{kj} \neq 0 \end{array} \right. \quad (64)$$

Under these assumptions Eq (60) becomes:

$$\begin{aligned} \Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\ &\quad + [(I - K_k H_k) \Phi_k - (I - K_j H_j) \Phi_k] \hat{x}_k(t_{i-1}') \\ &\quad + [(I - K_k H_k) B_k - (I - K_j H_j) B_k] u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') \\ &\quad + [(I - K_k H_k) - (I - K_j H_j)] \Phi_k \hat{x}_k(t_{i-1}') \\ &\quad + [(I - K_k H_k) - (I - K_j H_j)] B_k u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta K_{kj} z(t_i) \\ &\quad + [K_j H_j - K_k H_k] \Phi_k \hat{x}_k(t_{i-1}') + [K_j H_j - K_k H_k] B_k u(t_{i-1}) \\ &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta K_{kj} z(t_i) \\ &\quad + (K_j H_j - K_k H_k) [\Phi_k \hat{x}_k(t_{i-1}') + B_k u(t_{i-1})] \\ \Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta K_{kj} z(t_i) \\ &\quad + (K_j H_j - K_k H_k) \hat{x}_k(t_i') \end{aligned} \quad (65)$$

and Eq (58) becomes:

$$\begin{aligned} r_j(t_i) &= r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta H_{kj} \Phi_k \hat{x}_k(t_{i-1}') + \Delta H_{kj} B_k u(t_{i-1}) \\ &= r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta H_{kj} [\Phi_k \hat{x}_k(t_{i-1}') + B_k u(t_{i-1})] \\ r_j(t_i) &= r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta H_{kj} \hat{x}_k(t_i'). \end{aligned} \quad (66)$$

Returning to our flight control failure detection example, we find that we could construct the equivalent residuals of a bank of Kalman filters, each with a different sensor failure model, by using a single Kalman filter residual and Eqs (65) and (66). We will first develop the operations count assuming that  $\Delta K_{kj}$  is not necessarily sparse, and then develop the operations count assuming that  $\Delta K_{kj}$  is sparse with a small nonzero partition. We define  $n_H$  as the number of nonzero rows of  $\Delta H_{kj}$ . Since none of the terms in Eq (65) are sparse, the operations count for this equation is  $2n_s^2 + n_s n_m$  multiplications and  $2n_s^2 + n_s (n_m - 1)$  additions. The  $\Delta H_{kj}$  in Eq (66) is sparse, so only  $n_s n_H$  multiplies and  $n_H (n_s - 1)$  additions are need to evaluate the last term. Thus, the operations count for Eq (66) is  $n_s (n_H + n_m)$  multiplies and  $(n_H + n_m)(n_s - 1) + n_m$  additions. The total count for implementing the equivalent residual structure is  $n_s (2n_s + 2n_m + n_H)$  multiplies and  $n_s (2n_s + 2n_m + n_H) - (n_s - n_m + n_H)$  additions. This structure requires 1083 multiplies and 1072 additions for the specific application that we implemented, only because we assumed that  $\Delta K_{kj}$  is not necessarily sparse.

We observed that  $\Delta K_{kj}$  is actually sparse for the particular application that was implemented for this research. This matrix would have a small block partition that would be nonzero for the sensor failures that were modeled. We define the dimension of the block as  $n_K$ , and expanded the last term in Eq (65) to get  $\Delta K_{kj} \Delta H_{kj} - \Delta K_{kj} H_k - K_k \Delta H_{kj}$ . When the sparse structure of  $\Delta K_{kj}$  and  $\Delta H_{kj}$  are taken into account, this term is a sparse matrix with  $n_H$  nonzero rows and an  $n_K$  by  $n_K$  nonzero block partition. This results in a operations count of  $n_s + n_K (n_K - 1)$  multiplications and  $(n_s - 1) + (n_K - 1)(n_K - 1)$  additions. The total operations count for Eq (65) and Eq (66), when the sparse structures of  $\Delta K_{kj}$  and  $\Delta H_{kj}$  are taken into account, is  $n_s (n_s + n_m + n_H + 1) + n_K (2n_K - 1)$  multiplies and  $(n_s + n_m + n_H + 1)(n_s - 1) + (2n_K - 1)(n_K - 1) + 2n_m$  additions, which gives us 615 multiplies and 594 additions for this particular implementation where  $n_H = 1$  and  $n_K = 5$ . Subtracting this from the

operations count for the fully implemented Kalman filter version gives us a savings of  $n_s (n_m + n_i - n_H - 1) + n_H - 2n_K^2$  multiplies and  $n_s (n_m + n_i - n_H - 1) + n_H + 1 - (2n_K - 1)(n_K - 1) - n_m$ , which is a savings of 202 multiplies and 204 additions. Thus, the equivalent residual version of the Kalman filter bank requires about 25% fewer operations than the fully implemented Kalman filter bank.

### 3.2.5.5 Different State Transition Matrix Models. Finally, we

assume that the two Kalman filter models differ only in the state transition matrix. This will cause the two filters to have different Kalman filter gains, as explained in Section 3.2.2. This gives us:

$$\left. \begin{array}{l} \Phi_k \neq \Phi_j \rightarrow K_k \neq K_j \\ B_k = B_j \\ H_k = H_j \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \Delta \Phi_{kj} \neq 0 \\ \Delta B_{kj} = 0 \\ \Delta H_{kj} = 0 \\ \Delta K_{kj} \neq 0 \end{array} \right. \quad (67)$$

Under these assumptions Eq (60) becomes:

$$\begin{aligned} \Delta \epsilon_{jk}(t_i) &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}) \\ &\quad + [ (I - K_k H_k) \Phi_k - (I - K_j H_k) \Phi_j ] \hat{x}_k(t_{i-1}) \\ &\quad + [ (I - K_k H_k) B_k - (I - K_j H_k) B_k ] u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}) \\ &\quad + [ \Phi_k - \Phi_j - K_k H_k \Phi_k + K_j H_k \Phi_j ] \hat{x}_k(t_{i-1}) \\ &\quad + [ K_j H_k - K_k H_k ] B_k u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}) - \Delta K_{kj} H_k B_k u(t_{i-1}) + \Delta K_{kj} z(t_i) \\ &\quad + [ \Delta \Phi_{kj} - K_k H_k \Phi_k + K_j H_k (\Phi_k - \Delta \Phi_{kj}) ] \hat{x}_k(t_{i-1}) \end{aligned}$$

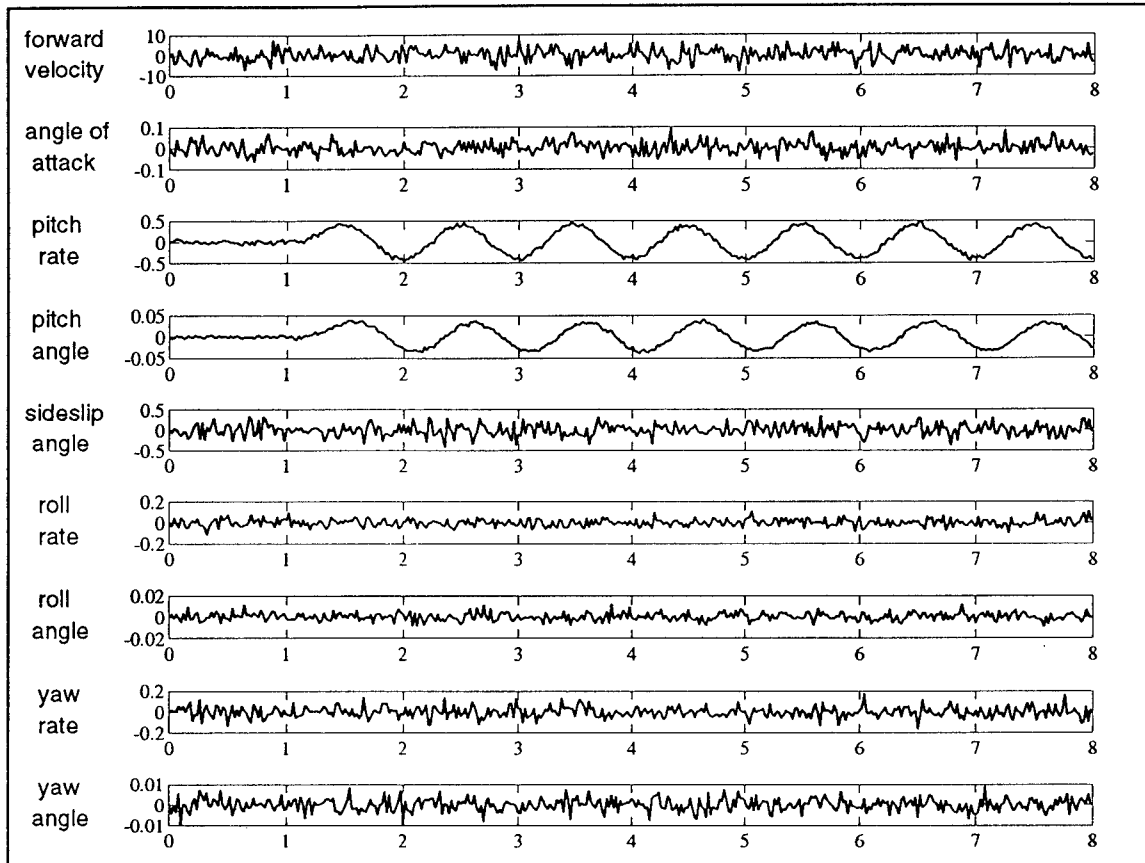
$$\begin{aligned}
\Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') - \Delta K_{kj} H_k B_k u(t_{i-1}) + \Delta K_{kj} z(t_i) \\
&\quad + [\Delta \Phi_{kj} - \Delta K_{kj} H_k \Phi_k - K_j H_k \Delta \Phi_{kj}] \hat{x}_k(t_{i-1}') \\
&= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta K_{kj} z(t_i) \\
&\quad + (I - K_j H_k) \Delta \Phi_{kj} \hat{x}_k(t_{i-1}') - \Delta K_{kj} H_k [\Phi_k \hat{x}_k(t_{i-1}') + B_k u(t_{i-1})] \\
&= (I - K_j H_j) \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + (I - K_j H_j) \Delta \Phi_{kj} \hat{x}_k(t_{i-1}') \\
&\quad + \Delta K_{kj} [z(t_i) - H_k \hat{x}_k(t_i)] \\
\Delta \epsilon_{jk}(t_i') &= (I - K_j H_j) [\Phi_j \Delta \epsilon_{jk}(t_{i-1}') + \Delta \Phi_{kj} \hat{x}_k(t_{i-1}')] + \Delta K_{kj} r_k(t_i)
\end{aligned} \tag{68}$$

and Eq (58) becomes:

$$r_j(t_i) = r_k(t_i) + H_j \Phi_j \Delta \epsilon_{jk}(t_{i-1}') + H_j \Delta \Phi_{kj} \hat{x}_k(t_{i-1}'). \tag{69}$$

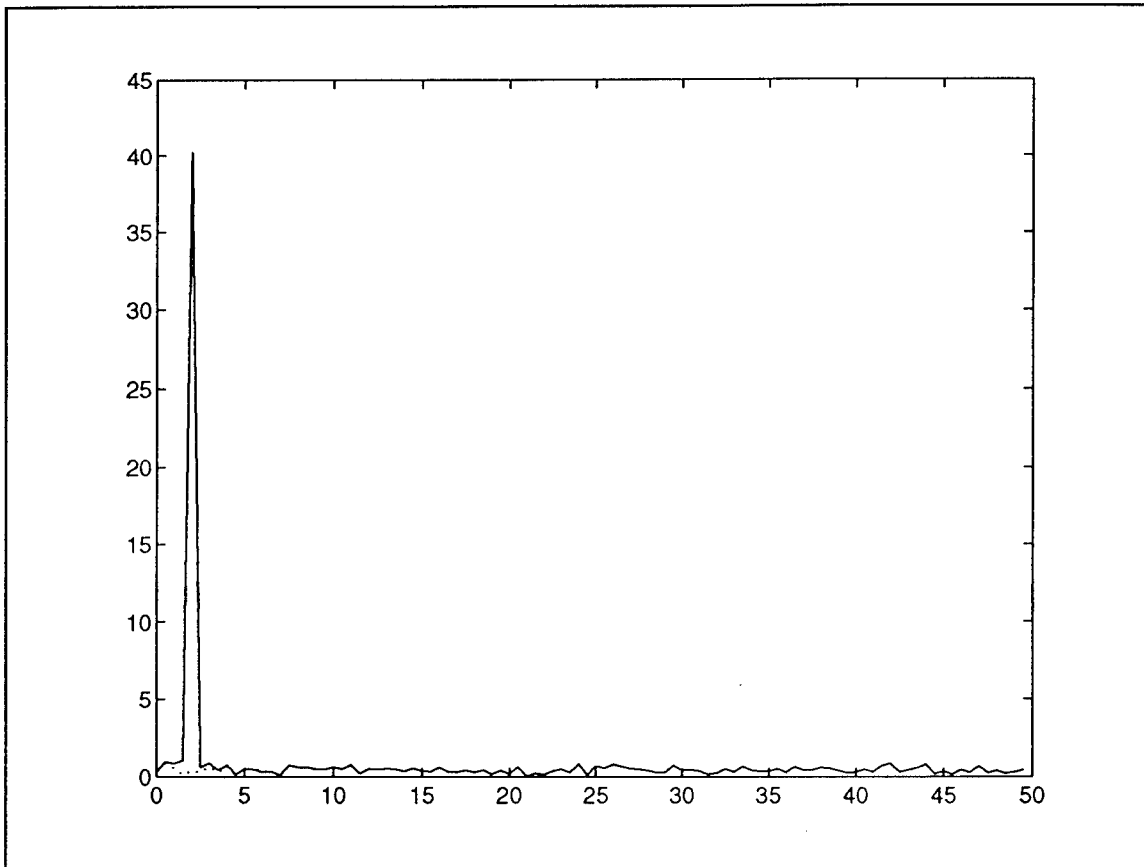
We can still construct the equivalent residuals of a bank of Kalman filters, each with a different state transition matrix model, by using a single Kalman filter residual and Eq (68) and Eq (69). The differences between models is sometimes in only one row and column of the state transition matrices, so  $\Delta \Phi_{kj}$  may be a matrix with only one nonzero row and column. When this is multiplied with another matrix, the product is no longer a sparse matrix. Thus, these equations will not, in general produce a savings in computational loading. However, just like the case of a mismodeled output matrix, the equivalent residual implementation may produce computational savings for some specific applications. Thus, the computational savings must be investigated for each specific application.

**3.2.6 Residual Correlation Kalman Filter Bank.** In this section we propose constructing a Kalman filter bank with outputs that are estimates of the power spectral density of each of the residuals from the Kalman filters in the bank. We showed earlier that, if the Kalman filter



**Figure 7.** Residual from the Kalman filter with the Fully Functional Model in the Presence of an Elevator Failure.

model is correct, the residual is a white sequence with zero mean, but if the model is incorrect the mean changes. This causes a change in the residual correlation, but not the covariance of the residual. In Section 3.2.4.2, we showed that, if the model differences are in the input matrix, the change in mean of the residual is a summation of input terms. If the input is a sinusoid, then these terms produce a sinusoidal residual at the same frequency as the input. This effect is clearly evident in Figure 7 where the residual from the Kalman filter based on a model that assumes a fully functional aircraft, shows the presence of the elevator dither input when an elevator failure occurs.



**Figure 8.** Fourier Transform of the Pitch Rate Residual from the Fully Functional Kalman Filter with Mismodeling (solid line) and No Mismodeling (dotted line).

Since we *know* the frequency of the input, we can use the spectral content of the residual at this particular frequency to indicate the presence of mismodeling. Figure 8 is the Fourier transform of 200 data points of the residual for the pitch rate residual element shown in Figure 7 (solid line), along with the same residual element from the same Kalman filter when there is no failure (dotted line). The "spike" in the solid line occurs at the elevator dither input frequency. Note that, at this particular frequency, the spectral content of the residual with the mismodeling is significantly greater than the spectral content for the correctly modeled residual. This figure shows that the spectral content of the residual clearly indicates the presence or absence of the mismodeling.

### 3.2.6.1 Basic Equations. Kay [26] develops several spectral estimation

techniques; we have chosen to use the periodogram [26: 65] for this research since its characteristics are well researched and it is used as a conceptual basis for many other spectral estimation techniques. One version of the periodogram utilizes the fast Fourier transform, which could greatly aid in implementing this technique because this transform has been implemented on commercial chip sets. Other spectral estimation techniques still need to be researched to determine which one or ones produce the desired performance. The periodogram is based on estimating the autocorrelation of the residual and then taking the discrete Fourier transform of the autocorrelation to produce an estimate of the power spectral density.

We must first make some assumptions to be able to estimate the autocorrelation of the residual. We assume that the residual sequence is a series of samples of a stationary process, so that the probability distributions do not change with time. We also assume that the residual is ergodic in the autocorrelation function, which implies that the expected value of the time-averaged autocorrelation function is the same irrespective of the length of time averaging.

For a multidimensional sequence, the estimate of the autocorrelation is:

$$\hat{A}_k(l) = \frac{1}{N} \sum_{n=0}^{N-1-|l|} \mathbf{r}_k(t_{i-n}) \mathbf{r}_k^T(t_{i-n-|l|}) \quad (70)$$

The periodogram is the discrete Fourier transform of this sequence, thus:

$$\begin{aligned}
\hat{\Psi}_k(f; t_i) &= \sum_{l=-(N-1)}^{N-1} \hat{A}_k(l) \exp(j 2 \pi f l) \\
&= \sum_{l=-(N-1)}^{N-1} \frac{1}{N} \sum_{n=0}^{N-1-|l|} r_k(t_{i-n}) r_k^T(t_{i-n-|l|}) \exp(j 2 \pi f l)
\end{aligned} \tag{71}$$

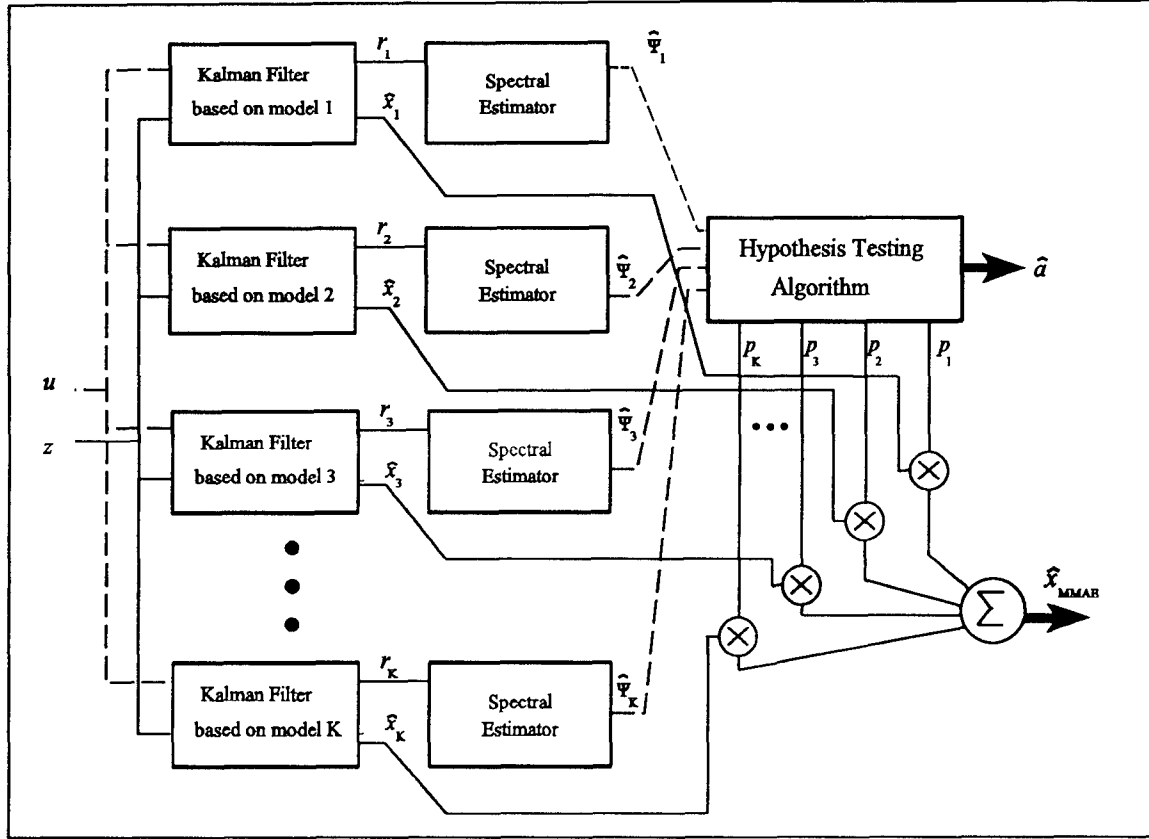
where  $N$  is the number of data samples that are collected over time.

Note that by dividing by  $N$  instead of  $(N-1)$  or  $(N-|l|-1)$  in Eq (70), we are using a biased estimate of the autocorrelation. We need the  $1/N$  factor to make this an estimate of the power spectral density.

If the residual sequence is a scalar sequence, than Eq (71) can be shown to be equivalent to:

$$\hat{\Psi}_k(f; t_i) = \frac{1}{N} \left| \sum_{n=0}^{N-1} r_k(t_{i-n}) \exp(j 2 \pi f l) \right|^2 \tag{72}$$

which is simply the squared absolute value of the  $N$ -point Fourier transform of the residual sequence. This is much easier to implement and executes much faster than Eq (71), since it exploits the fast Fourier transform routines. If the residual sequence is multidimensional, we can approximate Eq (71) with Eq (72) if the cross-correlation terms between elements of the residual vector are negligible. For the specific flight control application that we are studying, most of these cross terms are negligible. Since we are attempting to find practical implementations of the MMAE algorithm, and Eq (71) is computationally intensive, we chose to implement Eq (72) for this research, but we will continue the theoretical development based on Eq (71), since it is correct for multidimensional residual sequences.



**Figure 9.** Multiple Model Adaptive Estimation Algorithm using a Residual Correlation Kalman Filter Bank

We propose altering the Kalman filter bank by estimating the spectral content of each of the residuals using Eq (72). This structure is shown in Figure 9 (the "Hypothesis Testing Algorithm" block will be specified in detail in Section 3.3.1). Kay [26: 65-66] shows that this can be interpreted as filtering the residual with a bandpass filter centered at  $f$  and a 3dB bandwidth of  $1/N$ , sampling the output, and computing the squared magnitude. The  $1/N$  factor is need to make the estimate a power spectral density. Clearly, as more data samples are used ( $N$  increases) the filter bandwidth narrows and more of the out-of-bandwidth noise is rejected. To use this estimate of the power spectral density in the hypothesis testing algorithms that are developed in Section 3.3, we need the mean and covariance of this estimate.

Note that in Eqs (71) and (72) we are multiplying two Gaussian random variables together, which will yield a chi-squared distributed random variable, and then summing several of the chi-squared random variables together (we used  $N = 100$  for this research). Since each sample of the residual is independent and identically distributed, we can use the Lindeburg-Levy Theorem [53] to show that the distribution of  $\hat{\Psi}$  converges in distribution to a normal distribution. We generated histograms of  $\hat{\Psi}$  using 120 data points and observed that, for the case where the residual has a nonzero mean (when a failure occurs), this approximation works well. However, when the residual is zero-mean, then  $\hat{\Psi}$  appears to be more chi-squared distributed. We also observed that, for the dither input levels that we used for this research, the change in mean effects of  $\hat{\Psi}$  were much greater than the effect of using an incorrect distribution. In Section 3.3.1.1, we will approximate  $\hat{\Psi}$  as Gaussian. Further research needs to be accomplished to characterize the distribution of  $\hat{\Psi}$  properly, for a small number of data samples.

3.2.6.2 Mean of the Power Spectral Density Estimate. To develop the mean of this estimate, we first need to compute the autocorrelation function of the residual. Earlier, we found that, if there is no mismodeling, the autocorrelation matrix is:

$$\begin{aligned} A_k(l) &= \begin{cases} A_k(0), & \text{for } l = 0 \\ 0, & \text{for } l \neq 0 \end{cases} \\ &= A_k(0) \delta(l) \end{aligned} \quad (73)$$

where  $A_k(0) = H_k P_k H_k^T + R_k$ , and  $\delta(l)$  is the Kronecker delta function defined as:

$$\delta(l) = \begin{cases} 1, & \text{for } l = 0 \\ 0, & \text{for } l \neq 0 \end{cases} \quad (74)$$

We will now allow the mismodeling that we studied in the earlier sections and develop an expression for the residual autocorrelation.

We start by expanding the definition of the residual (letting  $v(t_i) = v_T(t_i)$ , since the subscript holds no information for this application):

$$\begin{aligned}
r_k(t_i) &= z(t_i) - H_k \hat{x}_k(t_i^-) \\
&= [H_T x_T(t_i) + v(t_i)] - H_k \hat{x}_k(t_i^-) \\
&= H_T x_T(t_i) - (H_T - \Delta H_k) \hat{x}_k(t_i^-) + v(t_i) \\
&= H_T [x_T(t_i) - \hat{x}_k(t_i^-)] + \Delta H_k \hat{x}_k(t_i^-) + v(t_i) \\
r_k(t_i) &= H_T \epsilon_k(t_i^-) + \Delta H_k \hat{x}_k(t_i^-) + v(t_i)
\end{aligned} \tag{75}$$

We assume that the mismodeling has occurred over the entire length of data sampling, all the way back to  $t_{i-l}$ , where  $l$  is the number of sample periods over which we are interested in estimating the autocorrelation. Since the residual is a real-valued sequence, we need only estimate the autocorrelation for positive  $l$  because the autocorrelation estimate for positive  $l$  is the same as for negative  $l$  ( $A_k(l) = A_k(-l)$ ). Now we use the definition of the autocorrelation of the residual, for  $l \geq 0$ :

$$\begin{aligned}
A_k(l) &= E_{Z(t_{i-1})} \{ r_k(t_i) r_k^T(t_{i-l}) \} \\
&= E_{Z(t_{i-1})} \left\{ \left[ H_T \epsilon_k(t_i^-) + \Delta H_k \hat{x}_k(t_i^-) + v(t_i) \right] \left[ \epsilon_k^T(t_{i-l}^-) H_T^T + \hat{x}_k^T(t_{i-l}^-) \Delta H_k^T + v^T(t_{i-l}) \right] \right\} \\
&= H_T E_{Z(t_{i-1})} \{ \epsilon_k(t_i^-) \epsilon_k^T(t_{i-l}^-) \} H_T^T + H_T E_{Z(t_{i-1})} \{ \epsilon_k(t_i^-) \hat{x}_k^T(t_{i-l}^-) \} \Delta H_k^T \\
&\quad + H_T E_{Z(t_{i-1})} \{ \epsilon_k(t_i^-) v^T(t_{i-l}) \} + \Delta H_k E_{Z(t_{i-1})} \{ \hat{x}_k(t_i^-) \epsilon_k^T(t_{i-l}^-) \} H_T^T \\
&\quad + \Delta H_k E_{Z(t_{i-1})} \{ \hat{x}_k(t_i^-) \hat{x}_k^T(t_{i-l}^-) \} \Delta H_k^T + \Delta H_k E_{Z(t_{i-1})} \{ \hat{x}_k(t_i^-) v^T(t_{i-l}) \} \\
&\quad + E_{Z(t_{i-1})} \{ v(t_i) \} E_{Z(t_{i-1})} \{ \epsilon_k^T(t_{i-l}^-) \} H_T^T + E_{Z(t_{i-1})} \{ v(t_i) \} E_{Z(t_{i-1})} \{ \hat{x}_k^T(t_{i-l}^-) \} \Delta H_k^T \\
&\quad + E_{Z(t_{i-1})} \{ v(t_i) v^T(t_{i-l}) \}
\end{aligned} \tag{76}$$

Using Eq (3) for the last three terms we get:

$$\begin{aligned}
E_{Z(t_{i-1})} \{v(t_i)\} E_{Z(t_{i-1})} \{\epsilon_k^T(t_{i-1}^-)\} H_T^T &= 0 \\
E_{Z(t_{i-1})} \{v(t_i)\} E_{Z(t_{i-1})} \{\hat{x}_k^T(t_{i-1}^-)\} \Delta H_T^T &= 0 \\
E_{Z(t_{i-1})} \{v(t_i) v^T(t_{i-1})\} &= R_T \delta(l)
\end{aligned} \tag{77}$$

The middle three terms are easily evaluated:

$$\begin{aligned}
\Delta H_k E_{Z(t_{i-1})} \{\hat{x}_k(t_i^-) \epsilon_k^T(t_{i-1}^-)\} H_T^T &= \Delta H_k \hat{x}_k(t_i^-) E_{Z(t_{i-1})} \{\epsilon_k^T(t_{i-1}^-)\} H_T^T \\
\Delta H_k E_{Z(t_{i-1})} \{\hat{x}_k(t_i^-) \hat{x}_k^T(t_{i-1}^-)\} \Delta H_T^T &= \Delta H_k \hat{x}_k(t_i^-) \hat{x}_k^T(t_{i-1}^-) \Delta H_T^T \\
\Delta H_k E_{Z(t_{i-1})} \{\hat{x}_k(t_i^-) v^T(t_{i-1})\} &= \Delta H_k \hat{x}_k(t_i^-) E_{Z(t_{i-1})} \{v^T(t_{i-1})\} = 0
\end{aligned} \tag{78}$$

The first three terms require an expression of  $\epsilon_k(t_i^-)$  in terms of  $\epsilon_k(t_{i-1}^-)$ . We will use a similar development to what we did in Section 3.2.4 where we derived an expression for  $\epsilon_k(t_i^+)$ , but we need  $\epsilon_k(t_i^-)$  for this expression. We start with the definition of  $\epsilon_k(t_i^-)$  and decrement one time sample:

$$\begin{aligned}
\epsilon_k(t_i^-) &= x_T(t_i) - \hat{x}_k(t_i^-) \\
&= [\Phi_T x_T(t_{i-1}) + B_T u(t_{i-1}) + G_T w(t_{i-1})] - [\Phi_k \hat{x}_k(t_{i-1}^-) + B_k u(t_{i-1})] \\
&= \Phi_T x_T(t_{i-1}) - (\Phi_T - \Delta \Phi_k) \hat{x}_k(t_{i-1}^-) + [B_T - (B_T - \Delta B_k)] u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T [x_T(t_{i-1}) - \hat{x}_k(t_{i-1}^-)] + \Delta \Phi_k \hat{x}_k(t_{i-1}^-) + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T [x_T(t_{i-1}) - \hat{x}_k(t_{i-1}^-) - K_k r_k(t_{i-1})] + \Delta \Phi_k [\hat{x}_k(t_{i-1}^-) + K_k r_k(t_{i-1})] \\
&\quad + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T \epsilon_k(t_{i-1}^-) - \Phi_T K_k [z(t_{i-1}) - H_k \hat{x}_k(t_{i-1}^-)] + \Delta \Phi_k \hat{x}_k(t_{i-1}^-) \\
&\quad + \Delta \Phi_k K_k [z(t_{i-1}) - H_k \hat{x}_k(t_{i-1}^-)] + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T \epsilon_k(t_{i-1}^-) - \Phi_T K_k [H_T x_T(t_{i-1}) + v(t_{i-1}) - (H_T - \Delta H_k) \hat{x}_k(t_{i-1}^-)] + \Delta \Phi_k \hat{x}_k(t_{i-1}^-) \\
&\quad + \Delta \Phi_k K_k [H_T x_T(t_{i-1}) + v(t_{i-1}) - (H_T - \Delta H_k) \hat{x}_k(t_{i-1}^-)] + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T \epsilon_k(t_{i-1}^-) - \Phi_T K_k H_T [x_T(t_{i-1}) - \hat{x}_k(t_{i-1}^-)] - \Phi_T K_k v(t_{i-1}) - \Phi_T K_k \Delta H_k \hat{x}_k(t_{i-1}^-) \\
&\quad + \Delta \Phi_k \hat{x}_k(t_{i-1}^-) + \Delta \Phi_k K_k H_T [x_T(t_{i-1}) - \hat{x}_k(t_{i-1}^-)] + \Delta \Phi_k K_k v(t_{i-1}) \\
&\quad + \Delta \Phi_k K_k \Delta H_k \hat{x}_k(t_{i-1}^-) + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1}) \\
&= \Phi_T (I - K_k H_T) \epsilon_k(t_{i-1}^-) - (\Phi_T - \Delta \Phi_k) K_k v(t_{i-1}) - (\Phi_T - \Delta \Phi_k) K_k \Delta H_k \hat{x}_k(t_{i-1}^-) \\
&\quad + \Delta \Phi_k \hat{x}_k(t_{i-1}^-) + \Delta \Phi_k K_k H_T \epsilon_k(t_{i-1}^-) + \Delta B_k u(t_{i-1}) + G_T w(t_{i-1})
\end{aligned}$$

$$\begin{aligned}\epsilon_k(t_i^-) = & \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right] \epsilon_k(t_{i-1}^-) - \Phi_k K_k v(t_{i-1}) + G_T w(t_{i-1}) \\ & + \left[ \Delta \Phi_k - \Phi_k K_k \Delta H_k \right] \hat{x}_k(t_{i-1}^-) + \Delta B_k u(t_{i-1})\end{aligned}\quad (79)$$

We apply this equation repeatedly to get:

$$\begin{aligned}\epsilon_k(t_i^-) = & \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l \epsilon_k(t_{i-l}^-) \\ & + \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H_k) \hat{x}_k(t_{i-1-n}^-) + \Delta B_k u(t_{i-1-n}) \right] \\ & + \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ G_T w(t_{i-1-n}) - \Phi_k K_k v(t_{i-1-n}) \right]\end{aligned}\quad (80)$$

Note that we have separated the noise terms from the deterministic terms in preparation of taking the expected value of this equation.

Now to use this in the first term of Eq (76):

$$\begin{aligned}H_T E_{Z(t_{i-1})} \left\{ \epsilon_k(t_i^-) \epsilon_k^T(t_{i-l}^-) \right\} H_T^T = & H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-l}^-) \epsilon_k^T(t_{i-l}^-) \right\} H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H_k) E_{Z(t_{i-1})} \left\{ \hat{x}_k(t_{i-1-n}^-) \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ \Delta B_k E_{Z(t_{i-1})} \left\{ u(t_{i-1-n}) \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ G_T E_{Z(t_{i-1})} \left\{ w(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ -\Phi_k K_k E_{Z(t_{i-1})} \left\{ v(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ = & H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-l}^-) \epsilon_k^T(t_{i-l}^-) \right\} H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H_k) \hat{x}_k(t_{i-1-n}^-) E_{Z(t_{i-1})} \left\{ \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ \Delta B_k u(t_{i-1-n}) E_{Z(t_{i-1})} \left\{ \epsilon_k^T(t_{i-l}^-) \right\} \right] H_T^T \\ = & H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-l}^-) \epsilon_k^T(t_{i-l}^-) \right\} H_T^T \\ & + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H_k) \hat{x}_k(t_{i-1-n}^-) + \Delta B_k u(t_{i-1-n}) \right] \\ & \cdot E_{Z(t_{i-1})} \left\{ \epsilon_k^T(t_{i-l}^-) \right\} H_T^T\end{aligned}\quad (81)$$

Using the same procedure for the second term in Eq (76):

$$\begin{aligned}
& H_T E_{Z(t_{i-1})} \left\{ \epsilon_k(t_i^-) \dot{x}_k^T(t_{i-1}) \right\} \Delta H_T^T \\
&= H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}^-) \dot{x}_k^T(t_{i-1}) \right\} \Delta H_T^T \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \\
&\quad \quad \cdot \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H) E_{Z(t_{i-1})} \left\{ \dot{x}_k(t_{i-1-n}) \dot{x}_k^T(t_{i-1}) \right\} \right] \Delta H_T^T \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ \Delta B_k E_{Z(t_{i-1})} \left\{ u(t_{i-1-n}) \dot{x}_k^T(t_{i-1}) \right\} \right] \Delta H_T^T \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ G_T E_{Z(t_{i-1})} \left\{ w(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ \dot{x}_k^T(t_{i-1}) \right\} \right] \Delta H_T^T \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ -\Phi_k K_k E_{Z(t_{i-1})} \left\{ v(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ \dot{x}_k^T(t_{i-1}) \right\} \right] \Delta H_T^T \\
&= H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}^-) \right\} \dot{x}_k^T(t_{i-1}) \Delta H_T^T \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H) \dot{x}_k(t_{i-1-n}) + \Delta B_k u(t_{i-1-n}) \right] \\
&\quad \quad \cdot \dot{x}_k^T(t_{i-1}) \Delta H_T^T
\end{aligned} \tag{82}$$

Finally, we use Eq (79) to expand on the third term in Eq (76)

$$\begin{aligned}
& H_T E_{Z(t_{i-1})} \left\{ \epsilon_k(t_i^-) v^T(t_{i-1}) \right\} \\
&= H_T \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}^-) \right\} E_{Z(t_{i-1})} \left\{ v^T(t_{i-1}) \right\} \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ (\Delta \Phi_k - \Phi_k K_k \Delta H) \dot{x}_k(t_{i-1-n}) E_{Z(t_{i-1})} \left\{ v^T(t_{i-1}) \right\} \right] \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ \Delta B_k E_{Z(t_{i-1})} \left\{ u(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ v^T(t_{i-1}) \right\} \right] \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ G_T E_{Z(t_{i-1})} \left\{ w(t_{i-1-n}) \right\} E_{Z(t_{i-1})} \left\{ v^T(t_{i-1}) \right\} \right] \\
&\quad + H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ -\Phi_k K_k E_{Z(t_{i-1})} \left\{ v(t_{i-1-n}) \right\} v^T(t_{i-1}) \right] \\
&= H_T \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T \right]^n \left[ -\Phi_k K_k R_T \delta(n-l+1) \right]
\end{aligned} \tag{83}$$

Note that  $\delta(n-l+1)$  is only nonzero for  $n = l-1$ , therefore there is only one term in the summation of Eq (84) that is nonzero. Also, the summation is only valid for  $l-1 \geq 0$ , or  $l \geq 1$ . Thus:

$$H_T E_{Z(t_{i-1})} \{ \epsilon_k(t_i^-) v^T(t_{i-1}) \} = -H_T [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^{l-1} \Phi_k K_k R_T \quad (84)$$

We put Eq (77), Eq (78), Eq (81), Eq (82), and Eq (84) into Eq (76), to get an expression for the residual autocorrelation with mismodeling in the output matrix, the control input matrix, and the state transition matrix:

$$\begin{aligned} A_k(l) = & H_T [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^l E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \epsilon_k^T(t_{i-1}) \} H_T^T \\ & + H_T \sum_{n=0}^{l-1} [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^n [ ( \Delta \Phi_k - \Phi_k K_k \Delta H ) \hat{x}_k(t_{i-1-n}) + \Delta B_k u(t_{i-1-n}) ] \\ & \quad \cdot E_{Z(t_{i-1})} \{ \epsilon_k^T(t_{i-1}) \} H_T^T \\ & + H_T [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^l E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \hat{x}_k^T(t_{i-1}) \Delta H_T^T \\ & + H_T \sum_{n=0}^{l-1} [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^n [ ( \Delta \Phi_k - \Phi_k K_k \Delta H ) \hat{x}_k(t_{i-1-n}) + \Delta B_k u(t_{i-1-n}) ] \\ & \quad \cdot \hat{x}_k^T(t_{i-1}) \Delta H_T^T \\ & - H_T [ \Phi_T(I - K_k H_T) + \Delta \Phi_k K_k H_T ]^{l-1} \Phi_k K_k R_T + \Delta H_k \hat{x}_k(t_i^-) E_{Z(t_{i-1})} \{ \epsilon_k^T(t_{i-1}) \} H_T^T \\ & + \Delta H_k \hat{x}_k(t_i^-) \hat{x}_k^T(t_{i-1}) H_T^T + R_T \delta(l) \end{aligned} \quad (85)$$

This is a ponderous expression to evaluate because we have allowed for any mismodeling in the control input matrix, output matrix, and state transition matrix. Most mismodeling in the MMAE filter banks occurs in only one of these matrices. Research [11, 12, 19, 41, 42, 43, 47, 48, 49, 52, 58, 60, 61] has shown that the MMAE identifies flight control sensor failures quite well, so we will target our research to improve the MMAE performance in detecting actuator failures. Thus, we will limit the mismodeling for the rest of this development to a mismodeled control input matrix. With

this assumption, we get  $\Delta H = 0$  and  $\Delta \Phi = 0$ , which simplifies the previous equation for the residual autocorrelation to:

$$\begin{aligned}
 A_k(l) = & H_T \left\{ \left[ \Phi_T(I - K_k H_T) \right]^l E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}) \epsilon_k^T(t_{i-1}) \right\} \right. \\
 & + \sum_{n=0}^{l-1} \left[ \Phi_T(I - K_k H_T) \right]^n \left[ \Delta B u(t_{i-1-n}) E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}) \right\} \right] \left. \right\} H_T^T \\
 & - H_T \left[ \Phi_T(I - K_k H_T) \right]^{l-1} \Phi_k K_k R_T + R_T \delta(l)
 \end{aligned} \tag{86}$$

To complete this derivation, we need to develop an expression for the first term in Eq (86). We start with:

$$\begin{aligned}
 E_{Z(t_{i-1})} \left\{ \epsilon_k(t_{i-1}) \epsilon_k^T(t_{i-1}) \right\} &= E_{Z(t_{i-1})} \left\{ \left[ x_T(t_{i-1}) - \hat{x}_k(t_{i-1}) \right] \left[ x_T(t_{i-1}) - \hat{x}_k(t_{i-1}) \right]^T \right\} \\
 &= E_{Z(t_{i-1})} \left\{ \left[ x_T(t_{i-1}) - \hat{x}_T(t_{i-1}) + \hat{x}_T(t_{i-1}) - \hat{x}_k(t_{i-1}) \right] \left[ x_T(t_{i-1}) - \hat{x}_k(t_{i-1}) \right]^T \right\} \\
 &= E_{Z(t_{i-1})} \left\{ \left[ x_T(t_{i-1}) - \hat{x}_T(t_{i-1}) + \Delta \epsilon_k(t_{i-1}) \right] \left[ x_T(t_{i-1}) - \hat{x}_k(t_{i-1}) \right]^T \right\} \\
 &= E_{Z(t_{i-1})} \left\{ \left[ \epsilon_T(t_{i-1}) + \Delta \epsilon_k(t_{i-1}) \right] \left[ \epsilon_T(t_{i-1}) + \Delta \epsilon_k(t_{i-1}) \right]^T \right\} \\
 &= E_{Z(t_{i-1})} \left\{ \epsilon_T(t_{i-1}) \epsilon_T^T(t_{i-1}) + E_{Z(t_{i-1})} \left\{ \epsilon_T(t_{i-1}) \Delta \epsilon_k(t_{i-1}) \right\} \right. \\
 &\quad \left. E_{Z(t_{i-1})} \left\{ \Delta \epsilon_k(t_{i-1}) \epsilon_T^T(t_{i-1}) \right\} + E_{Z(t_{i-1})} \left\{ \Delta \epsilon_k(t_{i-1}) \Delta \epsilon_k^T(t_{i-1}) \right\} \right\} \\
 &= P_T(t_{i-1}) + E_{Z(t_{i-1})} \left\{ \epsilon_T(t_{i-1}) \right\} \Delta \epsilon_k(t_{i-1}) + \Delta \epsilon_k(t_{i-1}) E_{Z(t_{i-1})} \left\{ \epsilon_T(t_{i-1}) \right\} \\
 &\quad + \Delta \epsilon_k(t_{i-1}) \Delta \epsilon_k^T(t_{i-1}) \\
 &= P_T + \Delta \epsilon_k(t_{i-1}) \Delta \epsilon_k^T(t_{i-1})
 \end{aligned} \tag{87}$$

where Eq (27) is used to define  $P_T(t_{i-1})$ , and we replace this with the steady state covariance matrix  $P_T$  in the last line of Eq (87). We can use this expression in Eq (86) to compute an estimate of the autocorrelation of the residual given a mismatching in the input matrix, and then use Eq (86) to compute the mean of the power spectral density estimate using:

$$\begin{aligned}
E_{Z(t_{i-1})} \{ \hat{\Psi}_k(f; t_i) \} &= \sum_{l=-(N-1)}^{N-1} E_{Z(t_{i-1})} \{ \hat{A}_k(l) \} \exp(-j2\pi fl) \\
&= \sum_{l=-(N-1)}^{N-1} \frac{1}{N} \sum_{n=0}^{N-1-|l|} E_{Z(t_{i-1})} \{ r_k(t_{i-n}) r_k^T(t_{i-n}, |l|) \} \exp(-j2\pi fl) \\
&= \sum_{l=-(N-1)}^{N-1} \frac{1}{N} \sum_{n=0}^{N-1-|l|} A_k(l) \exp(-j2\pi fl) \\
&= \sum_{l=-(N-1)}^{N-1} \frac{N-|l|}{N} A_k(l) \exp(-j2\pi fl) \\
&= A_k(0) + \sum_{l=1}^{N-1} \frac{N-l}{N} A_k(l) \exp(-j2\pi fl) + \sum_{l=-(N-1)}^{-1} \frac{N+|l|}{N} A_k(-l) \exp(-j2\pi fl) \\
&= A_k(0) + \sum_{l=1}^{N-1} \frac{N-l}{N} A_k(l) \exp(-j2\pi fl) + \sum_{m=1}^{N-1} \frac{N-m}{N} A_k(m) \exp(j2\pi fm) \\
&= A_k(0) + \sum_{l=1}^{N-1} \frac{N-l}{N} A_k(l) [\exp(-j2\pi fl) + \exp(j2\pi fl)] \\
E_{Z(t_{i-1})} \{ \hat{\Psi}_k(f; t_i) \} &= A_k(0) + 2 \sum_{l=1}^{N-1} \frac{N-l}{N} A_k(l) \cos(2\pi fl)
\end{aligned} \tag{88}$$

where we can use the relation  $A_k(l) = A_k(-l)$  since the residual is a real sequence. We can use Eq (86) and Eq (87) to compute the residual autocorrelation matrix and put this into Eq (88) to get the desired mean of the power spectral density estimate.

A more practical method of computing the mean of the power spectral density estimate would be to compute the mean of the residual, given the mismodeling in the Kalman filter, using the development in Section 3.2.4, and then compute the power spectral density estimate using Eq (71) or Eq (72). This method will not yield the correct mean of the spectral density estimate, but for large residuals it will provide an acceptable approximation.

### 3.2.6.3 Covariance of the Power Spectral Density Estimate. To

develop an expression for the covariance of the power spectral density estimate, we propose a

vectorized version of the power spectral density estimate matrix defined in Eq (71). To form this vector, we take the columns of the  $\hat{\Psi}$  matrix and stack them one on top of the other. Therefore:

$$\psi(f; t_i) = \begin{bmatrix} \hat{\Psi}_{1,1}(f; t_i) \\ \hat{\Psi}_{2,1}(f; t_i) \\ \hat{\Psi}_{3,1}(f; t_i) \\ \vdots \\ \hat{\Psi}_{m,m}(f; t_i) \end{bmatrix} \Rightarrow \psi(f; t_i)_a = \hat{\Psi}_{a - \left(\left\lceil \frac{a}{m} \right\rceil - 1\right) \cdot m, \left\lceil \frac{a}{m} \right\rceil}(f; t_i) \quad (89)$$

where  $m$  is the number of elements in the residual vector and  $\lceil \cdot \rceil$  is the ceiling function, which rounds the operand to the nearest integer toward positive infinity. Thus,  $\psi(f; t_i)$  is an  $m^2$ -dimensional vector of functions of frequency  $f$ . This construct will aid in notation for the covariance of this estimate since the covariance will be a matrix of scalars using this vector version, versus a tensor if we use the matrix version of the power spectral density estimate.

Using this notation, we will derive an expression for the autocorrelation of the power spectral density estimate. We will derive this expression for a single element of the autocorrelation matrix using  $a$  and  $b$  to denote that particular element (note that  $a$  and  $b$  can each take on integer values from 1 to  $m^2$ ). Also, we are denoting a particular element of the residual vector using  $j, k, l$ , or  $n$  as the index.

$$\begin{aligned}
& E_{Z(t_{-1})} \{ \psi(f; t_i) \psi^T(f; t_i) \}_{a,b} \\
&= E_{Z(t_{-1})} \left\{ \left[ \sum_{q=-(N-1)}^{N-1} \frac{1}{N} \sum_{s=0}^{N-1-|q|} r_j(t_{i-s}) r_k(t_{i-s-|q|}) \exp(-j2\pi f q) \right] \right. \\
&\quad \cdot \left. \left[ \sum_{p=-(N-1)}^{N-1} \frac{1}{N} \sum_{c=0}^{N-1-|p|} r_l(t_{i-c}) r_n(t_{i-c-|p|}) \exp(-j2\pi f p) \right] \right\} \\
&= \frac{1}{N^2} \sum_{q=-(N-1)}^{N-1} \sum_{p=-(N-1)}^{N-1} \sum_{s=0}^{N-1-|q|} \sum_{c=0}^{N-1-|p|} E_{Z(t_{-1})} \{ r_j(t_{i-s}) r_k(t_{i-s-|q|}) r_l(t_{i-c}) r_n(t_{i-c-|p|}) \} \\
&\quad \cdot \exp(-j2\pi f(p+q))
\end{aligned}$$

$$\text{where } j = \left\lceil \frac{a}{m} \right\rceil, k = a - (j-1)m, l = \left\lceil \frac{b}{m} \right\rceil, n = b - (l-1)m \quad (90)$$

This expression involves the fourth moment of a Gaussian random variable, which is difficult to expand if we allow any mismodeling in the Kalman filter. We intend to use this residual correlation Kalman filter bank with the standard hypothesis testing algorithm that is described in Section 3.3.1. This algorithm only requires the covariance of the power spectral density estimate for the case of no mismodeling in the Kalman filter. We showed earlier that this produces a zero-mean residual and we can compute the steady state covariance of the residual. Therefore, we can use the standard development of the fourth moment of a zero-mean Gaussian random variable [e.g. 37: 107] to expand this last expression, since the fourth moment can be expressed in terms of the second moment, which is the covariance of the residual. Thus:

$$\begin{aligned}
& E_{Z(t_{-1})} \{ \psi(f; t_i) \psi^T(f; t_i) \}_{a,b} \\
&= \frac{1}{N^2} \sum_{q=-(N-1)}^{N-1} \sum_{p=-(N-1)}^{N-1} \sum_{s=0}^{N-1-|q|} \sum_{t=0}^{N-1-|p|} [A_{j,k}(|q|) A_{l,n}(|p|) \\
&\quad + A_{j,l}(s-t) A_{k,n}(s+|q|-t-|p|) + A_{j,n}(s-t-|p|) A_{k,l}(s+|p|-t)] \\
&\quad \cdot \exp(-j2\pi f(p+q))
\end{aligned} \quad (91)$$

where  $A_{j,l}(s-t)$  denotes the  $j, l$  th element of the  $A_T(s-t)$  matrix.

This expression is very computationally intensive. Note that we are computing the covariance of the power spectral density estimate for the case where there is no mismodeling, so

$$A_T(t) = \begin{cases} A_k(0), & \text{for } t = 0 \\ \mathbf{0}, & \text{for } t \neq 0 \end{cases} \quad (92)$$

where  $A_k(0)$  is the precomputed steady state Kalman filter residual covariance matrix for the true Kalman filter. Thus, we can approximate Eq (91) by checking each of the time arguments of the residual covariance matrices and assume that the residual covariance matrix is  $\mathbf{0}$  if the time argument is not zero.

Earlier we showed that, for the case of a mismodeled input matrix, corresponding to an actuator failure, the residual covariance matrices are all equal. Thus, the covariance of the spectral density estimate will be the same for any of the actuator failure hypotheses.

### 3.3 Hypothesis Testing Algorithm

In Section 3.2 we showed that the Kalman filter residuals within an MMAE structure are normally distributed random vectors with known (and precomputable) steady state covariances, but different means. Therefore, the residuals will have different distributions that are dependent on the hypotheses of their internal models. The various hypotheses that were explored for this research are defined by  $\Delta H_k$ ,  $\Delta B_k$ , and  $\Delta \Phi_k$ . We will denote a set of these parameters as  $\theta$  and the parameter space of all possible parameter variations as  $\Theta$ . A particular hypothesis is constructed by defining a subset of the parameter space:

$$h_k: \theta \in \Theta_k \quad (93)$$

We also assume that the various hypotheses form a disjoint covering of the parameter space, so that

$$\Theta = \Theta_0 \cup \Theta_1 \cup \dots \cup \Theta_{N-1} \quad (94)$$

If we test  $h_0$  versus  $h_1$  versus ...  $h_{N-1}$ , then we have a N-ary hypothesis test. If  $N=2$  we have a binary hypothesis test. The primary hypothesis, which is the assumed true hypothesis at the time of the test, will be denoted  $h_0$ . The other hypotheses will be called the alternate hypotheses. Thus, a binary hypothesis test is a test of the primary hypothesis,  $h_0$ , versus the alternative hypothesis,  $h_1$ . N-ary hypothesis testing is an extension of binary hypothesis testing by simply making N-1 binary hypothesis tests between the primary hypothesis,  $h_0$ , and the N-1 alternate hypotheses,  $h_1$  through  $h_{N-1}$ , to obtain the desired N-ary hypothesis test. This is sufficient because the covering in Eq (94) is assumed to be disjoint. If each subspace  $\Theta_k$  contains a single element (i.e. represents a single flight control failure), then the hypothesis

$$h_k: \theta \in \Theta_k \quad (95)$$

is called a simple hypothesis. If it contains more than one element, then it is called a composite hypothesis. We will assume that the hypotheses we are researching are simple hypotheses (each subspace contains only a single flight control failure hypothesis) and denote a particular hypothesis as:

$$h_k: \theta = \Theta_k \quad (96)$$

There are two types of errors that can occur when testing hypotheses. Type I errors, called false alarms, occur when the alternate hypothesis is chosen when the true hypothesis is the primary hypothesis. The probability that this type of error can occur is called either the probability of false

alarm, denoted  $P_{FA}$ , or the size of the test, denoted  $\alpha$ . Type II errors, called missed detection, occur when the primary hypothesis is chosen when the true hypothesis is the alternate hypothesis. The probability that this type of error can occur is called the probability of a miss. This probability is  $1 - P_D$ , where  $P_D$  is the probability of detection, also called the power of the test.

Please note, the definitions for a false alarm and a missed detection that are used for Neyman-Pearson hypothesis testing. Researchers in the failure identification field usually define a false alarm as a declaration of a failure, when in fact one had not occurred, and a missed detection as a declaration of no-failure, when in fact a failure had occurred. These definitions agree with the Neyman-Pearson definitions above, *if* the current hypothesis is the no-failure hypothesis. Disagreement occurs when a failure hypothesis is in force. For example, if the assumed primary hypothesis is a failed elevator, and the hypothesis test declares a no-failure when in fact the elevator failure remains true, the Neyman-Pearson definition would define this a false alarm but the usual terminology would define this as a missed detection. We choose to use the Neyman-Pearson definitions because they are defined, without ambiguity, by the design parameters  $P_{FA}$  and the  $P_D$ .

3.3.1 Standard Hypothesis Testing Algorithm. First, we will give an example to demonstrate the conceptual basis of the Standard Hypothesis Testing Algorithm (SHTA), and then we'll derive the specific equations of this algorithm. Let us assign Kalman filter 1 the fully functional aircraft model, and Kalman filter 2 the left elevator failure model. Each of these filters would use the commanded input and its internal model to compute an estimate of what the measurements should be. Included in this computation is a statistical model of both the noise in these measurements and the disturbances that drive the system dynamics. The Kalman filter algorithm uses these models to calculate an expected value and covariance matrix of the measurement, before the actual measurement is taken at a particular sample time. The actual measurement is then used to form the

residual, which is the difference between the actual measurement and the computed prediction of that measurement. If the aircraft has no failures, then the residual from filter 1, whose internal model hypothesizes that there are no failures, would be much smaller, relative to its own internally computed covariance, than the residual in filter 2.

The SHTA would use the information in the residuals from the filter bank to compute the relative probabilities of each of the hypothesized models. The algorithm would assign the highest probability to filter 1 since it has the smallest residual. Now, let us assume that a left elevator failure occurs, thus the residual in filter 2 would become quite small and the residual in filter 1 would grow. The hypothesis testing algorithm would then assign less probability to the hypothesis of filter 1 and more to the hypothesis of filter 2.

#### 3.3.1.1 Basic Equations. The Standard Hypothesis Testing Algorithm

(SHTA) simultaneously tests the residuals of the Kalman filter bank under multiple hypotheses. In Section 3.2.4.1 we found that if the Kalman filter model matches the true system model, the residual has a mean of zero, and in Section 3.2.3 we found that the residual will have a precomputable covariance matrix,  $A_k$ . Therefore, this Kalman filter residual is a white Gaussian sequence of mean zero and covariance

$$A_k = H_k P_k^- H_k^T + R_k \quad (97)$$

Since we know that the residual is a Gaussian vector with zero mean and covariance  $A_k$ , we substitute these values into the known expression for a Gaussian conditional density function. Therefore we get that the conditional density function of the measurement ( $z$ ) at  $t_i$  for the  $k$ th Kalman filter, conditioned on the measurement history ( $Z(t_{i-1}) = [z^T(t_1) : \dots : z^T(t_{i-1})]^T$ ), is

$$f_z(t_i) | h_k, z(t_{i-1}) \quad (z_i | h_k, Z_{i-1}) = \beta_k \exp \{ \cdot \} \quad (98)$$

where  $\beta_k = \frac{1}{(2\pi)^{m/2} |A_k|^{1/2}}$  and  $\{ \cdot \} = \left\{ -\frac{1}{2} r_k^T(t_i) A_k^{-1} r_k(t_i) \right\}$

The scalar likelihood quotient will be defined as:

$$q_k(t_i) = r_k^T(t_i) A_k^{-1} r_k(t_i) \quad (99)$$

If the RCKFB, developed in Section 3.2.6, is implemented with the Standard Hypothesis Testing Algorithm (SHTA), we modify Eq (98) and Eq (99) to use the power spectral density estimate from the RCKFB. Two versions of the power spectral density estimate were developed in Section 3.2.6, each of which requires a slightly different modification to Eqs (98) and (99). The periodogram version, Eq (71), produces a matrix of power spectral density estimates because it included the cross power spectral density estimates between different elements of the residual. We computed the mean of this estimate, Eq (88) with  $A_k(l)$  given by Eq (85), and then formed a vector version of this matrix, Eq (89), to help in developing an expression for the covariance of the estimate, Eq (91). We introduced a simpler, less accurate version of the power spectral density estimate in Eq (72). This version produces a vector because it does not compute the cross power spectral density estimate between the elements of the residual, so it provides a subset of the estimates that are computed using Eq (71). The mean for this Fourier-transform-based version, can be computed either by using the appropriate terms of Eq (88), or by using the computation of the mean of the residual, developed in Section 3.2.4, in place of the residual in Eq (72). The covariance of the estimate can be computed using the appropriate terms of the covariance matrix from Eq (91).

Another method would be to take several samples of the power spectral density estimate, subtract the mean from the samples, and estimate the covariance of the samples. Many software packages, such as MATLAB, have an intrinsic sample covariance computation available, which makes this method quite easy to implement.

In practice, the mean for the no-failure hypothesis is very small when compared to the mean of the spectral density estimate for an actuator failure hypothesis. This is clearly seen in Figure 8, where the no-failure case has a mean of about 1, while the failed case has a mean of about 40. Since the disparity between the hypotheses is so great, we choose to model the power spectral density estimate as a zero-mean process with a covariance matrix that is assumed to be constant and computed using 400 samples of the fully functional residual power spectral density estimate. We evaluate  $\hat{\Psi}(f; t_i)$  at  $f = f_0$ , where  $f_0$  is the *known* frequency of the input. This gives us the modified version of Eq (98):

$$f_{\psi} | h_k, Z(t_{i-1}) \quad (\psi_k(f_0; t_i) | h, Z_{i-1}) = \beta_{\psi} \exp \{ \cdot \}$$

$$\text{where } \beta_{\psi} = \frac{1}{(2\pi)^{m^2/2} |A_{\psi}|^{1/2}} \quad \text{and } \{ \cdot \} = \left\{ -\frac{1}{2} \psi_k^T(f_0; t_i) A_{\psi}^{-1} \psi_k(f_0; t_i) \right\} \quad (100)$$

and Eq (99) becomes:

$$q_{\psi}(t_i) = \psi_k^T(f_0; t_i) A_{\psi}^{-1} \psi_k(f_0; t_i) \quad (101)$$

At this point, the development of the SHTA for the Standard Kalman Filter Bank (SKFB), Eqs (98) and (99), parallels the development of the SHTA for the RCKFB, Eqs (100) and (101). We present modifications to the SHTA for the SKFB, with the understanding that the same development would be used for the SHTA for the RCKFB, including the " $\beta$  Stripping" modification discussed in Section 3.3.1.2, which may create numerical conditioning problems due to the high dimensionality of  $\psi$ .

We can use these equations as a relative measure of the accuracy of the Kalman filter model. In Section 3.2.4 we found that if the Kalman filter model has some mismodeling errors, the mean of the residual will be nonzero, and if the Kalman filter model is accurate, the mean of the residual is zero. The Kalman filter that is most accurate will have a scalar likelihood quotient (the norm of the residual, scaled by the inverse of the precomputed covariance matrix) that is smaller than the scalar likelihood quotient from a Kalman filter with mismodeling errors. Thus, Eq (99) becomes a quadratic penalty for any mismodeling in the Kalman filter model. Conversely, if we use the scalar likelihood quotient as a negative exponent, as in the "dot" term of Eq (98), we will have an exponential indicator of the accuracy of the Kalman filter model. Note that if the residual is larger than expected due to modeling errors, the scalar likelihood quotient is large and the "dot" term will be small. If the "dot" term is scaled by the  $\beta_k$  term, we have a conditional probability density.

We define the conditional probability for a particular hypothesis as:

$$p_k(t_i) = \text{prob} \{ \mathbf{h} = \mathbf{h}_k \mid \mathbf{Z}(t_i) = \mathbf{Z}_i \} \quad (102)$$

We can compute the conditional probability for a particular hypothesis by comparing the conditional probability density for the current measurement, assuming that particular hypothesis, to the densities associated with all the other hypotheses. The conditional probability of a specific hypothesis can be shown [34] to be:

$$p_k(t_i) = \frac{f_z(t_i) \mid \mathbf{h}, \mathbf{Z}(t_{i-1}) \quad (\mathbf{z}_i \mid \mathbf{h}_k, \mathbf{Z}_{i-1}) \cdot p_k(t_{i-1})}{\sum_{j=1}^K f_z(t_i) \mid \mathbf{h}, \mathbf{Z}(t_{i-1}) \quad (\mathbf{z}_i \mid \mathbf{h}_j, \mathbf{Z}_{i-1}) \cdot p_j(t_{i-1})} \quad (103)$$

In this equation we use the prior conditional probabilities,  $p_k(t_{i-1})$ , to weigh the conditional probabilities of the current measurements, assuming each hypothesis, and then normalize it over the

complete set of such numerator terms. For failure identification applications, where we usually want to choose the most likely hypothesis out of the set of possible hypotheses, we would choose the hypothesis with the largest conditional probability. These conditional probabilities can also be used to weight and blend the various hypotheses, depending on the particular application.

In practice, these conditional probabilities will fluctuate rapidly from one time sample to the next. If the hypothesis testing algorithm is choosing the hypothesis with the highest conditional probability, these fluctuations could cause momentary incorrect hypothesis declarations. To alleviate this phenomenon, the conditional probability calculations are modified (as described in the next few sections) and compared to a decision threshold. The decision threshold is defined such that a hypothesis is chosen only if its conditional probability is greater than the threshold. The threshold is set to avoid momentary incorrect hypothesis declarations while hopefully providing adequate hypothesis testing performance. The performance is usually measured by the time it takes to detect and declare the correct flight control failure for the failure identification application. Various modifications to the conditional probabilities are explained below.

3.3.1.2  $\beta$  Stripping. Stevens [43, 60] found that certain performance problems could be reduced by modifying Eq (98). He altered the conditional density function in Eq (98) by removing the  $\beta_k$  term, which was used to make the area under the density function equal to 1. Thus Eq (98) becomes:

$$P_{h_k, Z(t_{i-1})}(z_i) = \exp \{ \cdot \} \quad (104)$$

$$\text{where } \{ \cdot \} = \left\{ -\frac{1}{2} \mathbf{r}_k^T(t_i) \mathbf{A}_k^{-1} \mathbf{r}_k(t_i) \right\}$$

where  $P_{h_k, Z(t_{i-1})}(z_i)$  is the unnormalized conditional probability of hypothesis  $k$ . Eq (103) becomes:

$$p_k(t_i) = \frac{P_{h_k, Z(t_{i-1})}(z_i) \cdot p_k(t_{i-1})}{\sum_{j=1}^K P_{h_j, Z(t_{i-1})}(z_i) \cdot p_j(t_{i-1})} \quad (105)$$

If all the scalar likelihood quotients were approximately the same size for all elemental filters, Eq (98) and Eq (103) would put the highest probability on the elemental filters with the smallest  $|A_k|$  value. This is an inappropriate weighting since the size of  $|A_k|$  has nothing to do with the correctness of the hypothesis in matching the current real-world failure status. Since sensor failures exhibit themselves as a row of  $H$  going to zero, filters based on the hypothesis of a failed sensor tend to have smaller  $|A_k|$  values, and thus the MMAE will be prone to false alarms on sensor failures.

Eq (104) and Eq (105) function properly with the  $\beta_k$  term removed because the denominator in Eq (105) is the sum of all possible numerators, so the probabilities ( $p_k$ ) will still sum to 1 even if the area under each of the modified "densities" of Eq (104), which are the densities of Eq (98) with  $\beta_k$  removed, is no longer unity.

### 3.3.1.3 Lower Bounding of the Conditional Probabilities.

Previous research [42, 43, 52, 60, 61] also found that a lower bound needed to be placed on the probabilities. The purpose of the MMAE is to make quick and accurate failure identifications, and it was found that if some of the probabilities were allowed to get too small, it took a long time for the probability in the correct filter to grow when a change in the failure status occurred. This was due to the fact that the previous conditional probability,  $p_k(t_{i-1})$ , was so small for the new correct filter model and so large for the old correct model, that Eq (105) had to be iterated several times before the values would change significantly.

This method was developed as an ad hoc way to allow for time varying model parameters. If we assumed that the model parameters were not constant, we would need to set up a filter bank in which each filter was based on a particular time history of the model parameters. This causes a geometric growth of the filter bank. A number of researchers have pursued such MMAE designs, as particularly with Markov process models for probability variations in time [13, 14, 32, 33], but some means of pruning or merging branches in the resulting decision tree structure is needed to prevent the impractical growth of the filter bank. Lower bounding the conditional probabilities in Eq (105) allows a change in model parameters to be reflected in the conditional probabilities *without* such dramatic bank growth. Menke [47, 48, 49] and Stratton [61] found that 0.001 was a good lower bound on the conditional probabilities for an application using about ten elemental filters.

3.3.1.4 Filter Tuning. Previous experiments [19, 41] found that an incorrect hypothesis may be momentarily chosen because the scalar likelihood quotients representing the incorrect hypothesis may temporarily be smaller than the likelihood quotients from the correct hypothesis. One cause of this phenomenon was the norm of  $A_k$  for these incorrect hypotheses. These hypotheses had smaller values for the norm of  $A_k$ , and thus had smaller scalar likelihood quotients for a given vector residual. As noted in Section 3.3.1.2, the norm of  $A_k$  has nothing to do with the correctness of the hypothesis, so these small norm values were causing momentary incorrect hypothesis declarations.

This effect was corrected by tuning the Kalman filters that used the incorrect hypothesis models. The tuning was accomplished by increasing the modeled value of measurement noise,  $R$ , which is called adding measurement pseudonoise. Adding the measurement pseudonoise changes the Kalman filter gain so that when the state estimates are updated, the measurements are not weighted as heavily as they were before the pseudonoise was added. This technique is especially appropriate if

one of the scalar measurements is particularly noisy with respect to the other scalar measurements. Another method of increasing the norm of  $A_k$  is to increase the modeled value of the dynamic noise covariance,  $Q$ . This also changes the Kalman filter gain, but with this technique the measurements are weighted more heavily than the predictions of those measurements based on the Kalman filter model. If the filters are mistuned, the accuracy of the state estimates is much poorer than can be achieved with better tuning of the Kalman filter gain values. The appropriate amount of pseudonoise had to be found through extensive simulation to prevent performance degradation due to mistuning.

3.3.1.5 Smoothing of the Conditional Probabilities. The false alarm rate in some previous research [42, 43, 52, 60, 61] was found to be too high in most simulation examples. To decrease the false alarm rate, the conditional probabilities were averaged over several time samples. A set of  $N$  conditional probabilities was collected and then averaged using:

$$p_k(t_i) = \frac{1}{N} \sum_{j=0}^{N-1} p_j(t_{i-j}) \quad (106)$$

This average was then compared to the threshold and a hypothesis was chosen. This decreased the effects of momentary false alarms, but it also delayed the hypothesis identification.

3.3.1.6 Exponential Penalty Increase. The SHTA was found to be much slower in declaring the correct failure for certain failure hypotheses than for others. These failures took much longer to identify because the effects of the failure had to grow for some time before they were disambiguated from other failure hypotheses. To remedy this, a scalar multiplier was substituted into Eq (104) to get:

$$P_{h_k, z(t_{i-1})}(z_i) = \exp\{\cdot\} \quad \text{where} \quad \{\cdot\} = \left\{ -\Gamma r_k^T(t_i) A_k^{-1} r_k(t_i) \right\} \quad (107)$$

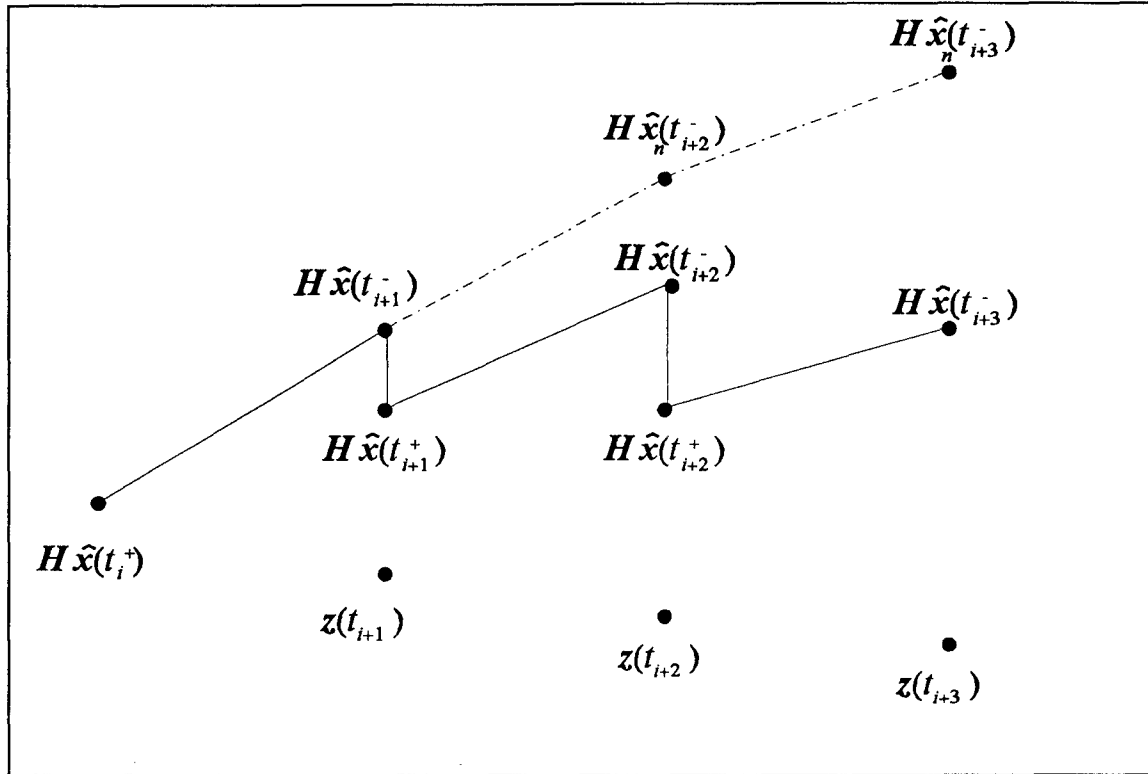
Research [19, 41] has shown that by increasing the exponential penalty for an incorrect hypothesis, that is increasing  $\Gamma$  in Eq (107), caused the SHTA to choose the correct hypothesis faster. However, the number of false alarms also increased when this penalty was increased. The penalty needed to produce the desired performance had to be found through extensive simulation.

### 3.3.1.7 Propagating Several Time Samples before Updating.

Another approach to rectifying the problem presented in the last section is to propagate the state estimate several time periods before updating. Figure 10 shows graphically how this approach would work with a scalar state estimate. Normally, the state estimates are updated at each time sample, which moves the state estimate closer to the measurement, as shown in Figure 10. If the state estimate is propagated without update for several time samples, the distance between the state estimate and the measurement will be much greater, if the hypothesized model used by the filter is incorrect. This is clear in Figure 10 where the  $\hat{x}_n$  state estimate is propagated without update for three iterations and the distance between  $H\hat{x}_n(t_{i+3})$  and  $z(t_{i+3})$  is much larger than the distance between  $H\hat{x}(t_{i+3})$  and  $z(t_{i+3})$ . This will produce a larger residual, which produces a larger penalty for mismodeling, thus helping to disambiguate the hypotheses. Others [20] have researched this approach.

## 3.3.2 Neyman-Pearson Based Hypothesis Testing Algorithm.

3.3.2.1 Basic Equations. In this section we will develop an alternative hypothesis testing algorithm based on the Neyman-Pearson Lemma. We will first define a binary hypothesis test and then present the Neyman-Pearson Lemma. Then we'll show that this alternative hypothesis testing algorithm requires only a single Kalman filter residual, instead of a complete bank of Kalman filters.



**Figure 10** - Comparison of Updated State Estimates with Nonupdated State Estimates.

A binary test of  $h_0$ :  $\theta \in \Theta_0$  (the parameter set is in the parameter subspace  $\Theta_0$ ) versus  $h_1$ :  $\theta \in \Theta_1$  of a residual would take the form

$$\phi \{r(t_i)\} = \begin{cases} 1 \sim h_1, & r \in R \\ 0 \sim h_0, & r \in A. \end{cases} \quad (108)$$

This is read "the test function  $\phi(r)$  equals 1, and hypothesis  $h_1$  is accepted, if the residual lies in the  $h_0$  rejection region  $R$ . The test function equals zero, and hypothesis  $h_0$  is accepted, if the residual lies in the  $h_0$  acceptance region  $A$ ."

Considering the flight control failure detection application, our desire is to define the acceptance and rejection regions such that the detection probability,  $P_D$ , is maximized for a given probability of false alarm,  $P_{FA}$ . We can usually accept a certain rate of false alarms, but, it is

essential not to miss detecting the failure with this particular application. There are, however, other applications that might need to minimize the probability of a false alarm for a given detection probability.

Now that we have defined the basic binary hypothesis test, we can introduce the Neyman-Pearson Lemma [53: 107]:

#### Neyman-Pearson Lemma

Let  $\Theta = \{ \theta_0, \theta_1 \}$  and assume that  $\mathbf{r}$  has a density or probability mass function, denoted  $f_{\theta}(\mathbf{r})$ . The subscript denotes that the function exists over the whole parameter space  $\Theta$ . This function will be subscripted with  $\theta_0$  or  $\theta_1$  to denote the function for a particular subset of the parameter space. The test of the form

$$\phi(\mathbf{r}) = \begin{cases} 1, & \text{if } f_{\theta_1}(\mathbf{r}) > k f_{\theta_0}(\mathbf{r}) \\ \zeta, & \text{if } f_{\theta_1}(\mathbf{r}) = k f_{\theta_0}(\mathbf{r}) \\ 0, & \text{if } f_{\theta_1}(\mathbf{r}) < k f_{\theta_0}(\mathbf{r}) \end{cases} \quad (109)$$

for some test threshold  $k \geq 0$  and some  $0 \leq \zeta \leq 1$ , is the most powerful test (the test with the largest  $P_D$ ) of size  $\alpha > 0$  (for a given  $P_{FA}$ ) for testing  $H_0: \theta = \theta_0$  versus  $H_1: \theta = \theta_1$ . These tests are unique except perhaps on sets of probability zero under  $H_0$  and  $H_1$ . When  $\phi(\mathbf{r}) = 1$  we choose  $H_1$ , and when  $\phi(\mathbf{r}) = 0$  we choose  $H_0$ . When  $\phi(\mathbf{r}) = \zeta$  we select  $H_1$  with probability  $\zeta$ .

This lemma can be rewritten by defining the likelihood ratio as:

$$I(\mathbf{r}) = \frac{f_{\theta_1}(\mathbf{r})}{f_{\theta_0}(\mathbf{r})} \quad (110)$$

so that the most powerful test of size  $\alpha$  for testing  $h_0: \theta = \theta_0$  versus  $h_1: \theta = \theta_1$  is a likelihood ratio test of the form

$$\phi \{ r \} = \begin{cases} 1, & \text{if } l(r) > k \\ \zeta, & \text{if } l(r) = k \\ 0, & \text{if } l(r) < k \end{cases} \quad (111)$$

If  $l(r) = k$  with probability zero, then  $\zeta = 0$  and the threshold  $k$ , for a given size  $\alpha$ , can be computed from:

$$\alpha = P_{\theta_0} [ l(r) > k ] = \int_k^{\infty} f_{\theta_0}(l) dl \quad (112)$$

where  $f_{\theta_0}(l)$  is the density function of  $l(r)$  under  $h_0$ . The power of the test,  $P_D$ , can be obtained from:

$$\Omega = P_D = \int_k^{\infty} f_{\theta_1}(l) dl. \quad (113)$$

We will now show that the likelihood ratio in Eq (110), which operates on a single residual, is equivalent to a likelihood ratio that operates on two different residuals. First, we assume that the density functions used in the likelihood ratio are conditioned on the measurement history  $Z(t_{i-1})$ . We will compare two residuals from two different Kalman filters,  $r_j$  and  $r_k$ . The Kalman filter models used in these filters use two different hypotheses that differ only in the control input matrix, output matrix, or dynamics matrix. We will compare the two likelihood ratios to check for equality:

$$\frac{f_{\theta_1}(r_j)}{f_{\theta_0}(r_k)} \stackrel{?}{=} \frac{f_{\theta_1}(r_j)}{f_{\theta_0}(r_j)}$$

$$\begin{aligned}
& \rightarrow \frac{\beta_1 \exp \left\{ -\frac{1}{2} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_1 \right]^T A_1^{-1} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_1 \right] \right\}}{\beta_0 \exp \left\{ -\frac{1}{2} \left[ \mathbf{r}_k - E_{Z(t_{i-1})} \{ \mathbf{r}_k \} \mid h_0 \right]^T A_0^{-1} \left[ \mathbf{r}_k - E_{Z(t_{i-1})} \{ \mathbf{r}_k \} \mid h_0 \right] \right\}} \\
& \stackrel{?}{=} \frac{\beta_1 \exp \left\{ -\frac{1}{2} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_1 \right]^T A_1^{-1} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_1 \right] \right\}}{\beta_0 \exp \left\{ -\frac{1}{2} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_0 \right]^T A_0^{-1} \left[ \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_0 \right] \right\}} \quad (114)
\end{aligned}$$

For simplicity of notation, we are suppressing the time argument,  $t_i$ , of the residuals. Examining Eq (114), we note that the numerators and leading coefficients are equal. Thus the equality will hold if and only if:

$$\mathbf{r}_k - E_{Z(t_{i-1})} \{ \mathbf{r}_k \} \mid h_0 \stackrel{?}{=} \mathbf{r}_j - E_{Z(t_{i-1})} \{ \mathbf{r}_j \} \mid h_0 \quad (115)$$

We assume that the hypothesis,  $h_0$ , can consist of any of the modeling differences that we are investigating, namely  $\Delta B$ ,  $\Delta H$ , and  $\Delta \Phi$ . We substitute Eq (35) and Eq (36) into Eq (115) and cancel terms to get:

$$\begin{aligned}
& H_T \Phi_T \left[ \epsilon_k(t_{i-1}) - E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \right] \cdot H_T G_T \mathbf{w}_d(t_{i-1}) + \mathbf{v}_T(t_i) \\
& \stackrel{?}{=} H_T \Phi_T \left[ \epsilon_j(t_{i-1}) - E_{Z(t_{i-1})} \{ \epsilon_j(t_{i-1}) \} \right] \cdot H_T G_T \mathbf{w}_d(t_{i-1}) + \mathbf{v}_T(t_i) \\
& \rightarrow \epsilon_k(t_{i-1}) - E_{Z(t_{i-1})} \{ \epsilon_k(t_{i-1}) \} \stackrel{?}{=} \epsilon_j(t_{i-1}) - E_{Z(t_{i-1})} \{ \epsilon_j(t_{i-1}) \} \quad (116)
\end{aligned}$$

We then expand these terms using the definition from Eq (23) to get:

$$\begin{aligned}
& \left[ \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_k(t_{i-1}) \right] - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_k(t_{i-1}) \} \\
& \stackrel{?}{=} \left[ \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_j(t_{i-1}) \right] - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_j(t_{i-1}) \} \\
& \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_k(t_{i-1}) - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) \} + \hat{\mathbf{x}}_k(t_{i-1}) \\
& \stackrel{?}{=} \mathbf{x}_T(t_{i-1}) - \hat{\mathbf{x}}_j(t_{i-1}) - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) \} + \hat{\mathbf{x}}_j(t_{i-1}) \\
& \mathbf{x}_T(t_{i-1}) - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) \} \stackrel{!}{=} \mathbf{x}_T(t_{i-1}) - E_{Z(t_{i-1})} \{ \mathbf{x}_T(t_{i-1}) \} \quad (117)
\end{aligned}$$

Therefore

$$\frac{f_{\theta_1}(r_j)}{f_{\theta_0}(r_k)} = \frac{f_{\theta_1}(r_j)}{f_{\theta_0}(r_j)} \quad (118)$$

Thus we can perform the hypothesis test using a single Kalman filter residual, testing it for multiple hypotheses, and this is equivalent to performing the hypothesis test using multiple residuals, each one tested for a single hypothesis.

### 3.3.2.2 Single Time Sample Hypothesis Testing. Now we can apply

the Neyman-Pearson hypothesis test to the residual of a Kalman filter. We need an estimate of the residual covariance for a particular hypothesis, which we can precompute, as described in Section 3.2.1, by using a Kalman filter that uses the hypothesized system model. We will use the steady state Kalman filter estimate of the residual covariance matrix, thus the residual covariance will be considered constant. Note that the estimate for the residual covariance is not a function of the system input matrix,  $B$ . Thus, if we only have modeling differences in the system input matrix, so only  $\Delta B$  is nonzero, then the residual covariance matrix will be identical for each of the hypotheses. We will develop the Neyman-Pearson based hypothesis testing algorithm for the case of  $\Delta B \neq 0$ , which is an actuator failure for the flight control application. Later we will look at the case where the other modeling differences are nonzero.

Under these assumptions, we have a Neyman-Pearson detector for "common covariances, uncommon means" as developed by Scharf [53: 111]. We will develop the Neyman-Pearson detector in the same manner as Scharf does, but we will first extend it to a single-time sample hypothesis test using a multidimensional random variable (a single time sample of the residual). In Section 3.3.2.3, we will extend the development to use several time samples of the residual.

We will start with a single time sample of a Kalman filter residual and design a binary test using the Neyman-Pearson Lemma. We first need to compute the likelihood ratio. We've shown that the residual is normally distributed with covariance  $A_T$  irrespective of the hypothesis that is currently in force in the detection algorithm. These residual covariances, which we will denote  $A_k$ , for the various hypotheses are identical only if the modeling differences are limited to  $\Delta B \neq 0$  and all other differences (i.e.,  $\Delta \Phi$  and  $\Delta H$ ) are zero. We will denote the mean of the residual given the past measurement history and a particular hypothesis as

$$\begin{aligned} E_{Z(t_{i-1})} \{ r(t_i) \} |_{h_0} &= m_0(t_i) \\ E_{Z(t_{i-1})} \{ r(t_i) \} |_{h_1} &= m_1(t_i). \end{aligned} \quad (119)$$

Using Eq (110), the likelihood ratio is a ratio of two normal density functions, which becomes the ratio of two exponential terms when the leading  $\beta$  terms are canceled. Thus the likelihood ratio becomes:

$$\begin{aligned} l(r(t_i)) &= \exp \left\{ -\frac{1}{2} [r(t_i) - m_1(t_i)]^T A_k^{-1} [r(t_i) - m_1(t_i)] \right. \\ &\quad \left. + \frac{1}{2} [r(t_i) - m_0(t_i)]^T A_k^{-1} [r(t_i) - m_0(t_i)] \right\} \\ &= \exp \left\{ -\frac{1}{2} [r(t_i)^T A_k^{-1} r(t_i) - r(t_i)^T A_k^{-1} m_1(t_i) - m_1(t_i)^T A_k^{-1} r(t_i) + m_1(t_i)^T A_k^{-1} m_1(t_i)] \right. \\ &\quad \left. + \frac{1}{2} [r(t_i)^T A_k^{-1} r(t_i) - r(t_i)^T A_k^{-1} m_0(t_i) - m_0(t_i)^T A_k^{-1} r(t_i) + m_0(t_i)^T A_k^{-1} m_0(t_i)] \right\} \\ &= \exp \left\{ \frac{1}{2} [r(t_i)^T A_k^{-1} m_1(t_i) + m_1(t_i)^T A_k^{-1} r(t_i) - m_1(t_i)^T A_k^{-1} m_1(t_i) \right. \\ &\quad \left. - r(t_i)^T A_k^{-1} m_0(t_i) - m_0(t_i)^T A_k^{-1} r(t_i) + m_0(t_i)^T A_k^{-1} m_0(t_i)] \right\}. \end{aligned} \quad (120)$$

Note that each term within the exponential is a scalar and we can use one of the properties of the covariance matrix  $A_k$  to get:

$$\begin{aligned}
(A_k^{-1})^T &= A_k^{-1} \\
r(t_i)^T A_k^{-1} m_1(t_i) &= \left( r(t_i)^T A_k^{-1} m_1(t_i) \right)^T = m_1(t_i)^T (A_k^{-1})^T r(t_i) = m_1(t_i)^T A_k^{-1} r(t_i) \\
r(t_i)^T A_k^{-1} m_0(t_i) &= \left( r(t_i)^T A_k^{-1} m_0(t_i) \right)^T = m_0(t_i)^T (A_k^{-1})^T r(t_i) = m_0(t_i)^T A_k^{-1} r(t_i) \quad (121)
\end{aligned}$$

Also, we can use the natural log of the likelihood ratio in place of the likelihood ratio because the natural logarithm is a monotonic function. Therefore we get the log likelihood ratio:

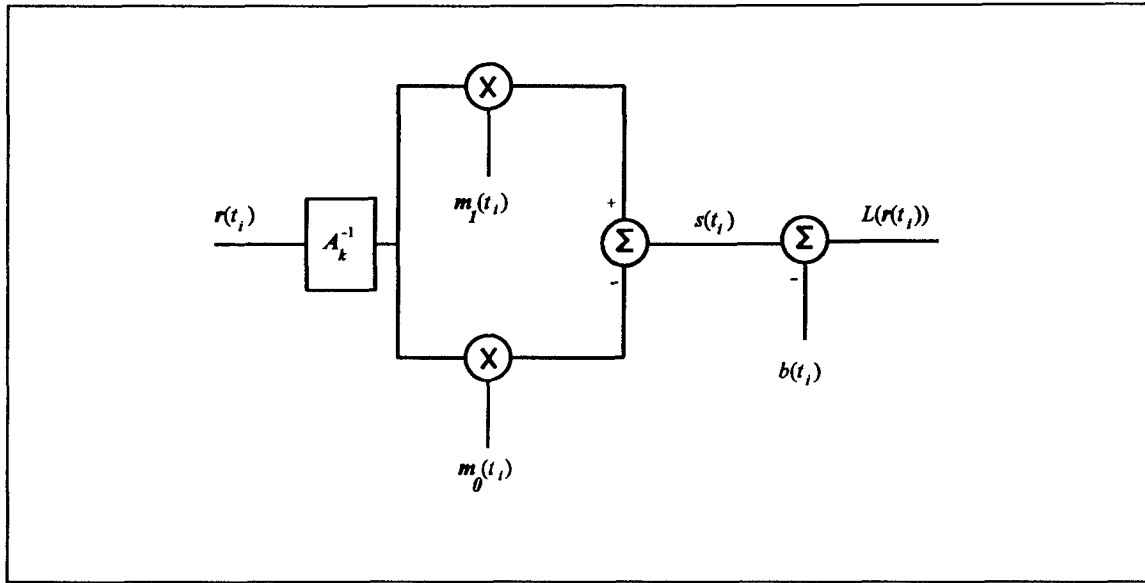
$$\begin{aligned}
L(r(t_i)) &= \ln \{ l(r(t_i)) \} \\
&= m_1(t_i)^T A_k^{-1} r(t_i) - m_0(t_i)^T A_k^{-1} r(t_i) - \frac{1}{2} \left( m_1(t_i)^T A_k^{-1} m_1(t_i) - m_0(t_i)^T A_k^{-1} m_0(t_i) \right) \\
&= (m_1(t_i) - m_0(t_i))^T A_k^{-1} r(t_i) - \frac{1}{2} (m_1(t_i) - m_0(t_i))^T A_k^{-1} (m_1(t_i) + m_0(t_i)) \\
&= s(t_i) - b(t_i)
\end{aligned}$$

$$\begin{aligned}
\text{where } s(t_i) &= (m_1(t_i) - m_0(t_i))^T A_k^{-1} r(t_i) \\
\text{and } b(t_i) &= \frac{1}{2} (m_1(t_i) - m_0(t_i))^T A_k^{-1} (m_1(t_i) + m_0(t_i)) \quad (122)
\end{aligned}$$

We call  $s(t_i)$  the signal and  $b(t_i)$  the bias of the log likelihood function. One structure for implementing Eq (122) is shown in Figure 11.

Rewriting the Neyman-Pearson Lemma to use the log likelihood function in Eq (122), we get:

$$\phi \{ r(t_i) \} = \begin{cases} 1, & L(r(t_i)) > \eta = \ln(k) \\ 0, & L(r(t_i)) \leq \eta \end{cases} \quad (123)$$



**Figure 11 - Implementation of the Neyman-Pearson Based Hypothesis Test**

Now we only need to compute the threshold  $\eta$  in order to define the Neyman-Pearson hypothesis test algorithm completely. We first need to find the probability density function for the test statistic  $L(r(t_i))$ . We note the  $L$  is a linear transformation of  $r$ , therefore  $L$  is also normally distributed and we need only compute the mean and covariance to define the probability density function. We'll start by computing the mean under both the primary and alternate hypotheses.

First we define:

$$d(t_i) = m_1(t_i) - m_0(t_i)$$

so that the test statistic becomes:

$$\begin{aligned} L(r(t_i)) &= d(t_i)^T A_k^{-1} r(t_i) - \frac{1}{2} d(t_i)^T A_k^{-1} (m_1(t_i) + m_0(t_i)) \\ &= s(t_i) - b(t_i). \end{aligned} \tag{124}$$

Now we compute the mean of the test statistic, given that the primary hypothesis is in force:

$$\begin{aligned}
E_{Z(t_{i-1})}\{L(r(t_i))\} |_{h_0} &= d(t_i)^T A_k^{-1} E_{Z(t_{i-1})}\{r(t_i)\} |_{h_0} - \frac{1}{2} d(t_i)^T A_k^{-1} (m_1(t_i) + m_0(t_i)) \\
&= d(t_i)^T A_k^{-1} m_0(t_i) - \frac{1}{2} d(t_i)^T A_k^{-1} m_1(t_i) - \frac{1}{2} d(t_i)^T A_k^{-1} m_0(t_i) \\
&= \frac{1}{2} d(t_i)^T A_k^{-1} (m_0(t_i) - m_1(t_i)) = -\frac{1}{2} d(t_i)^T A_k^{-1} d(t_i) = -\frac{1}{2} D(t_i)
\end{aligned} \tag{125}$$

where  $D(t_i)$  will be called the hypothesis discrimination measure.

Likewise, we compute the mean of the test statistic given that the alternative hypothesis is in force:

$$\begin{aligned}
E_{Z(t_{i-1})}\{L(r(t_i))\} |_{h_1} &= d(t_i)^T A_k^{-1} E_{Z(t_{i-1})}\{r(t_i)\} |_{h_1} - \frac{1}{2} d(t_i)^T A_k^{-1} (m_1(t_i) + m_0(t_i)) \\
&= d(t_i)^T A_k^{-1} m_1(t_i) - \frac{1}{2} d(t_i)^T A_k^{-1} m_1(t_i) - \frac{1}{2} d(t_i)^T A_k^{-1} m_0(t_i) \\
&= \frac{1}{2} d(t_i)^T A_k^{-1} (m_1(t_i) - m_0(t_i)) = \frac{1}{2} d(t_i)^T A_k^{-1} d(t_i) = \frac{1}{2} D(t_i).
\end{aligned} \tag{126}$$

Now to compute the conditional covariance of the test statistic under either hypothesis:

$$\begin{aligned}
cov_{Z(t_{i-1})}\{L(r(t_i))\} &= E_{Z(t_{i-1})}\{L(r(t_i))L(r(t_i))^T\} - E_{Z(t_{i-1})}\{L(r(t_i))\}E_{Z(t_{i-1})}\{L(r(t_i))^T\} \\
&= E_{Z(t_{i-1})}\{[s(t_i) - b(t_i)]^2\} - [E_{Z(t_{i-1})}\{s(t_i) - b(t_i)\}]^2 \\
&= E_{Z(t_{i-1})}\{s(t_i)^2 - 2s(t_i)b(t_i) + b(t_i)^2\} - [E_{Z(t_{i-1})}\{s(t_i)\} - b(t_i)]^2 \\
&= E_{Z(t_{i-1})}\{s(t_i)^2\} - 2E_{Z(t_{i-1})}\{s(t_i)\}b(t_i) + b(t_i)^2 \\
&\quad - [E_{Z(t_{i-1})}\{s(t_i)\}^2 - 2E_{Z(t_{i-1})}\{s(t_i)\}b(t_i) + b(t_i)^2] \\
&= E_{Z(t_{i-1})}\{s(t_i)^2\} - E_{Z(t_{i-1})}\{s(t_i)\}^2.
\end{aligned} \tag{127}$$

Now use Eq (124) to get:

$$\begin{aligned}
\text{cov}_{Z(t_{i-1})}\{L(r(t_i))\} &= E_{Z(t_{i-1})}\{s(t_i)^2\} - E_{Z(t_{i-1})}\{s(t_i)\}^2 \\
&= E_{Z(t_{i-1})}\left\{\left[d(t_i)^T A_k^{-1} r(t_i)\right]^2\right\} - E_{Z(t_{i-1})}\left\{\left[d(t_i)^T A_k^{-1} r(t_i)\right]\right\}^2 \\
&= E_{Z(t_{i-1})}\left\{\left[d(t_i)^T A_k^{-1} r(t_i)\right]\left[d(t_i)^T A_k^{-1} r(t_i)\right]^T\right\} \\
&\quad - E_{Z(t_{i-1})}\left\{\left[d(t_i)^T A_k^{-1} r(t_i)\right]\right\} E_{Z(t_{i-1})}\left\{\left[d(t_i)^T A_k^{-1} r(t_i)\right]^T\right\} \\
&= d(t_i)^T A_k^{-1} E_{Z(t_{i-1})}\{r(t_i) r(t_i)^T\} A_k^{-1} d(t_i) \\
&\quad - d(t_i)^T A_k^{-1} E_{Z(t_{i-1})}\{r(t_i)\} E_{Z(t_{i-1})}\{r(t_i)^T\} A_k^{-1} d(t_i) \\
&= d(t_i)^T A_k^{-1} \left[ E_{Z(t_{i-1})}\{r(t_i) r(t_i)^T\} - E_{Z(t_{i-1})}\{r(t_i)\} E_{Z(t_{i-1})}\{r(t_i)^T\} \right] A_k^{-1} d(t_i) \\
&= d(t_i)^T A_k^{-1} A_k A_k^{-1} d(t_i) \\
&= d(t_i)^T A_k^{-1} d(t_i) = D(t_i).
\end{aligned} \tag{128}$$

We assumed that the residual correlation matrix was identical for each of the hypotheses when we formed the likelihood ratio in Eq (120). This occurs when we only allow differences in the control input matrix ( $B$ ), which corresponds to actuator failures for flight control failure identification applications. We noted earlier, that model differences in the output matrix ( $H$ ) and state transition matrix ( $\Phi$ ) will produce different residual correlation matrices. We will now look at the case where we are modeling a flight control sensor failure.

A flight control sensor failure is modeled by zeroing a row of the output matrix that corresponds to the particular failed sensor. Thus, the element of the vector  $H \hat{x}(t_i)$  that corresponds to the failed sensor will be zero. Therefore, this element of the residual vector will primarily affect the row and column of the residual covariance matrix that corresponds to the failed sensor and the other elements of the covariance matrix may be slightly affected. It has been observed that the row and column of the residual covariance matrix that correspond to the failed sensor increase significantly, which is appropriate since this indicates that the Kalman filter algorithm based on such

a residual covariance matrix will not rely on the failed sensor measurement as much as it did before the failure. One approach to detecting flight control sensor failures would be to conduct a Neyman-Pearson hypothesis test, assuming that the residual correlation matrices are equal to the computed residual correlation matrix for the sensor failure (and correspondingly for the element of  $H \hat{x}(t_i)$  corresponding to the failed sensor to be set to zero) This approach will work well if the effect of the change in mean of the residual is much more significant than the effect of the difference between the correlation matrices. A similar argument can be made for some mismodeling in the state transition matrix. Therefore, we are assuming that the state estimates are large enough to produce a significant change in the residual mean and that the residual correlation matrices are approximately equal when we conduct a hypothesis test for a sensor failure, which is tantamount to assuming that the residual mean effects will dominate the effects of differing correlation matrices.

To summarize, we know that the test statistic is normally distributed as follows:

$$\begin{aligned} h_0 : L(r(t_i)) &\sim N\left[-\frac{D(t_i)}{2}, D(t_i)\right] \\ h_1 : L(r(t_i)) &\sim N\left[\frac{D(t_i)}{2}, D(t_i)\right]. \end{aligned} \quad (129)$$

We substitute this into the known density function for a normal distribution, which can then be used to compute the threshold  $\eta$  for a given  $P_{FA}$ . Using the definition given in Eq (110), we get:

$$\begin{aligned} \alpha = P_{FA} &= \int_{\eta}^{\infty} \frac{1}{(2\pi D(t_i))^{1/2}} \exp\left\{-\frac{1}{2} \frac{\left(x + \frac{D(t_i)}{2}\right)^2}{D(t_i)}\right\} dx = \int_{\left[\frac{\eta + \frac{D(t_i)}{2}}{D(t_i)^{1/2}}\right]}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp\left\{-\frac{y^2}{2}\right\} dy \\ &= 1 - \gamma\left(\frac{\eta + \frac{D(t_i)}{2}}{D(t_i)^{1/2}}\right) = 1 - \gamma(g) \end{aligned} \quad (130)$$

where

$$g = \frac{\eta + \frac{D(t_i)}{2}}{D(t_i)^{1/2}} \quad \gamma(g) = \int_{-\infty}^g \frac{1}{(2\pi)^{1/2}} \exp\left\{-\frac{x^2}{2}\right\} dx \quad \gamma(-g) = 1 - \gamma(g). \quad (131)$$

Similarly, we can compute the  $P_D$ :

$$\begin{aligned} P_D &= \int_{\eta}^{\infty} \frac{1}{(2\pi D(t_i))^{1/2}} \exp\left\{-\frac{1}{2} \frac{\left(x - \frac{D(t_i)}{2}\right)^2}{D(t_i)}\right\} dx = \int_{\left[\frac{\eta - \frac{D(t_i)}{2}}{D(t_i)^{1/2}}\right]}^{\infty} \frac{1}{(2\pi)^{1/2}} \exp\left\{-\frac{y^2}{2}\right\} dy \\ &= 1 - \gamma\left(\frac{\eta - \frac{D(t_i)}{2}}{D(t_i)^{1/2}}\right) = 1 - \gamma(g - D(t_i)^{1/2}). \end{aligned} \quad (132)$$

For most applications, the designer would set  $P_{FA}$  and  $P_D$  to achieve the desired hypothesis testing performance, and that would dictate a unique hypothesis discrimination measure,  $D(t_i)$ , needed to achieve the  $P_{FA}$  and  $P_D$ . However, this does not give us a unique  $d(t_i)$  needed to achieve the discrimination measure, even for a given  $A_k^{-1}$ . Let's assume that one of the hypotheses, say  $h_0$ , is correct, so that  $m_0(t_i) = 0$ . This gives us  $d(t_i) = m_1(t_i)$ , which we can use, along with the precomputable  $A_k^{-1}$ , to get a class of residual means that would achieve the desired discrimination measure. In practice, we would choose a system input, compute the residual mean, and compute the discrimination measure to see it achieves the desired value. If it doesn't, we iterate on the input until the desired discrimination measure is achieved.

The process for computing the required  $D(t_i)$  is summarized at the end of the next section. Unfortunately, for the flight control application that we have been considering, a single time sample of the residual will not produce enough distinguishability between two hypotheses. We therefore will look at using multiple samples of the residual to perform the hypothesis testing.

### 3.3.2.3 Multiple Time Samples Hypothesis Testing. We propose

simply to sum several time samples of the log likelihood test statistic when the hypothesis discrimination is too small to attain the desired hypothesis testing performance. Therefore, let

$$\mathcal{L}(r(t_i)) = \sum_{n=0}^{N-1} L(r(t_{i-n})). \quad (133)$$

We now need to compute the mean of this new test statistic under each hypothesis. To compute the conditional mean of this test statistic, we will use the conditional mean of each of the terms in the summation of Eq (133), conditioned on the measurement history up to the last measurement before the log likelihood test statistic,  $L(r(t_{i-n}))$ , is computed. This will allow us to compute the log likelihood test statistic at each time sample, and add it to a running sum of previous log likelihood test statistics, which is precisely what we want to implement in real time. To denote the different conditional means on each of the log likelihood test statistics, we will define:

$$\begin{aligned} E_{Z(t_{i-2})}\{\mathcal{L}(r(t_i))\} &= \sum_{n=0}^{N-1} E_{Z(t_{i-1-n})}\{L(r(t_{i-n}))\} \\ &= E_{Z(t_{i-1})}\{L(r(t_i))\} + E_{Z(t_{i-2})}\{L(r(t_{i-1}))\} + \dots + E_{Z(t_{i-N+2})}\{L(r(t_{i-N+1}))\}. \end{aligned} \quad (134)$$

With this notation, we get:

$$\begin{aligned} E_{Z(t_{i-2})}\{\mathcal{L}(r(t_i))\} |_{h_0} &= \sum_{n=0}^{N-1} E_{Z(t_{i-1-n})}\{L(r(t_{i-n}))\} |_{h_0} = \sum_{n=0}^{N-1} \left(-\frac{1}{2}\right) D(t_{i-n}) \\ &= -\frac{1}{2} \sum_{n=0}^{N-1} D(t_{i-n}) \\ E_{Z(t_{i-2})}\{\mathcal{L}(r(t_i))\} |_{h_1} &= \sum_{n=0}^{N-1} E_{Z(t_{i-1-n})}\{L(r(t_{i-n}))\} |_{h_1} = \sum_{n=0}^{N-1} \left(\frac{1}{2}\right) D(t_{i-n}) \\ &= \frac{1}{2} \sum_{n=0}^{N-1} D(t_{i-n}). \end{aligned} \quad (135)$$

Now to compute the conditional covariance of this test statistic. As we did above, we will condition the expectation operator on the measurement history up to the latest data sample in the summation that we are operating on. Therefore:

$$\text{cov}_{Z(t_{-N})}\{\mathcal{Q}(r(t_i))\} = E_{Z(t_{-N})}\{\mathcal{Q}(r(t_i))^2\} - \left[ E_{Z(t_{-N})}\{\mathcal{Q}(r(t_i))\} \right]^2. \quad (136)$$

Since we are developing a recursive relationship of  $\mathcal{Q}(r(t_i))$  that is based on the previous values, we will separate the sum as follows:

$$\begin{aligned} \text{cov}_{Z(t_{-N})}\{\mathcal{Q}(r(t_i))\} &= E_{Z(t_{-N})}\left\{\left[\sum_{n=0}^{N-1} L(r(t_{i-n}))\right]^2\right\} - \left[E_{Z(t_{-N})}\left\{\sum_{n=0}^{N-1} L(r(t_{i-n}))\right\}\right]^2 \\ &= E_{Z(t_{-N})}\left\{\left[L(r(t_i)) + \sum_{n=1}^{N-1} L(r(t_{i-n}))\right]^2\right\} \\ &\quad - \left[E_{Z(t_{-N})}\left\{L(r(t_i)) + \sum_{n=1}^{N-1} L(r(t_{i-n}))\right\}\right]^2. \end{aligned} \quad (137)$$

Expanding this relationship we get:

$$\begin{aligned} &\text{cov}_{Z(t_{-N})}\{\mathcal{Q}(r(t_i))\} \\ &= E_{Z(t_{-N})}\left\{L(r(t_i))^2 + 2L(r(t_i))\sum_{n=1}^{N-1} L(r(t_{i-n})) + \left[\sum_{n=1}^{N-1} L(r(t_{i-n}))\right]^2\right\} \\ &\quad - \left[E_{Z(t_{-N})}\{L(r(t_i))\} + E_{Z(t_{-N})}\left\{\sum_{n=1}^{N-1} L(r(t_{i-n}))\right\}\right]^2 \\ &= E_{Z(t_{-N})}\{L(r(t_i))^2\} + 2E_{Z(t_{-N})}\left\{L(r(t_i))\sum_{n=1}^{N-1} L(r(t_{i-n}))\right\} \\ &\quad + E_{Z(t_{-N})}\left\{\left[\sum_{n=1}^{N-1} L(r(t_{i-n}))\right]^2\right\} - \left[E_{Z(t_{-N})}\{L(r(t_i))\}\right]^2 \\ &\quad - 2E_{Z(t_{-N})}\{L(r(t_i))\}\left[E_{Z(t_{-N})}\left\{\sum_{n=1}^{N-1} L(r(t_{i-n}))\right\}\right] - \left[E_{Z(t_{-N})}\left\{\sum_{n=1}^{N-1} L(r(t_{i-n}))\right\}\right]^2 \end{aligned}$$

$$\begin{aligned}
& cov_{Z(t_{i-2})} \{ \mathcal{L}(r(t_i)) \} \\
&= \left[ E_{Z(t_{i-2})} \{ L(r(t_i))^2 \} - \left( E_{Z(t_{i-2})} \{ L(r(t_i)) \} \right)^2 \right] \\
&+ 2 \left[ E_{Z(t_{i-2})} \left\{ L(r(t_i)) \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} - E_{Z(t_{i-2})} \{ L(r(t_i)) \} E_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \right] \\
&+ \left[ E_{Z(t_{i-2})} \left\{ \left( \sum_{n=1}^{N-1} L(r(t_{i-n})) \right)^2 \right\} - \left( E_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \right)^2 \right] \\
\therefore cov_{Z(t_{i-2})} \{ \mathcal{L}(r(t_i)) \} &= cov_{Z(t_{i-2})} \{ L(r(t_i)) \} + cov_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \\
&+ 2 cov_{Z(t_{i-2})} \left\{ L(r(t_i)), \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\}. \tag{138}
\end{aligned}$$

If we assume that the first two terms of Eq (138) dominate the third term, so that:

$$\begin{aligned}
& cov_{Z(t_{i-2})} \{ L(r(t_i)) \} + cov_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \\
&\gg 2 cov_{Z(t_{i-2})} \left\{ L(r(t_i)), \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \tag{139}
\end{aligned}$$

then, using this approximation, we get:

$$\begin{aligned}
cov_{Z(t_{i-2})} \{ \mathcal{L}(r(t_i)) \} &\approx cov_{Z(t_{i-2})} \{ L(r(t_i)) \} + cov_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \\
&\approx cov_{Z(t_{i-1})} \{ L(r(t_i)) \} + cov_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \\
&\approx D(t_i) + cov_{Z(t_{i-2})} \left\{ \sum_{n=1}^{N-1} L(r(t_{i-n})) \right\} \tag{140}
\end{aligned}$$

Since this holds for any  $N$ , we use this repeatedly to get:

$$\begin{aligned}
\text{cov}_{Z(t_{i-1})}\{\mathcal{L}(r(t_i))\} &\approx D(t_i) + \text{cov}_{Z(t_{i-2})}\{L(r(t_{i-1}))\} + \dots + \text{cov}_{Z(t_{i-N})}\{L(r(t_{i-N+1}))\} \\
&\approx D(t_i) + D(t_{i-1}) + \dots + D(t_{i-N+1}) \\
&\approx \sum_{n=0}^{N-1} D(t_{i-n}).
\end{aligned} \tag{141}$$

Thus we can approximate the covariance of the test statistic by recursively adding the covariances at each time sample. We will define the discrimination measure for the test statistic,  $\mathcal{L}(r(t_i))$ , as:

$$\Delta(t_i) = \sum_{n=0}^{N-1} D(t_{i-n}). \tag{142}$$

We have now characterized the test statistic,  $\mathcal{L}(r(t_i))$ , proposed in Eq (133) by defining the mean of the test statistic, Eq (135), under both the primary and alternative hypotheses, and an approximation of the covariance of the test statistic, Eq (142). These equations were defined for a given number of data samples,  $N$ , but what defines the required number of data samples? We will now address this question.

The performance of the hypothesis test is defined by the probability of false alarms,  $P_{FA}$ , and the probability of detection,  $P_D$ . If we can compute the inverse of the  $\gamma$  function defined in Eq (131), then we can compute the test threshold and the desired discrimination measure. We will denote the desired discrimination measure as  $\Delta_T$ , and call it the trigger discrimination measure for reasons which will become clear shortly. We will denote the test threshold as  $\eta$ , as before, and the inverse of the  $\gamma$  function as  $\gamma^{-1}$ . We will begin the computation of the test threshold and trigger discrimination measure by using:

$$\begin{aligned}
P_{FA} &= 1 - \gamma\left(\frac{\eta + \frac{\Delta_T}{2}}{\Delta_T^{1/2}}\right) = 1 - \gamma(g) \\
\rightarrow g &= \gamma^{-1}(1 - P_{FA})
\end{aligned} \tag{143}$$

and

$$\begin{aligned}
 P_D &= 1 - \gamma \left( \frac{\eta - \frac{\Delta_T}{2}}{\Delta_T^{1/2}} \right) = 1 - \gamma (g - \Delta_T^{1/2}) \\
 \rightarrow g - \Delta_T^{1/2} &= \gamma^{-1} (1 - P_D).
 \end{aligned} \tag{144}$$

Subtracting Eq (144) from Eq (143), we get:

$$\begin{aligned}
 g - (g - \Delta_T^{1/2}) &= \Delta_T^{1/2} = \gamma^{-1} (1 - P_{FA}) - \gamma^{-1} (1 - P_D) \\
 \rightarrow \Delta_T &= [\gamma^{-1} (1 - P_{FA}) - \gamma^{-1} (1 - P_D)]^2.
 \end{aligned} \tag{145}$$

The test threshold can be computed using the definition of  $g$ :

$$\begin{aligned}
 g &= \frac{\eta \cdot \frac{\Delta_T}{2}}{\sqrt{\Delta_T}} \\
 \rightarrow \eta &= \sqrt{\Delta_T} \gamma^{-1} (1 - P_{FA}) - \frac{\Delta_T}{2}.
 \end{aligned} \tag{146}$$

Some software packages (such as MATLAB) do not have the inverse of the  $\gamma$  function available, but they usually have an inverse error function available. We will extend the usual development [53: 114-115], Eqs (143) through (146), to the case where we do not have the inverse of the  $\gamma$  function available. We can use the following relations

$$\begin{aligned}
 1 - \text{erf}(y) &= \text{erfc}(y) = \frac{2}{\sqrt{\pi}} \int_y^\infty \exp(-t^2) dt = 2 \gamma(\sqrt{2} y) \\
 \text{erfinv}(x) &= y \rightarrow x = \text{erf}(y)
 \end{aligned} \tag{147}$$

to compute the trigger discrimination measure and the test threshold.

First, we compute the test threshold using:

$$\begin{aligned}
 \text{let } \eta &= \sqrt{2} \, y \rightarrow y = \frac{\eta}{\sqrt{2}} \\
 \rightarrow 2 \, \gamma(\eta) &= 2 \, \gamma(\sqrt{2} \, y) = 2 \, P_{FA} = 1 - \text{erf}(y) \\
 \rightarrow \text{erf}\left(\frac{\eta}{\sqrt{2}}\right) &= 1 - 2 \, P_{FA} \rightarrow \text{erfinv}(1 - 2 \, P_{FA}) = \frac{\eta}{\sqrt{2}} \\
 \rightarrow \eta &= \sqrt{2} \, \text{erfinv}(1 - 2 \, P_{FA}).
 \end{aligned} \tag{148}$$

Then we can compute the trigger discrimination measure using:

$$\begin{aligned}
 \text{let } \eta - \Delta_T &= \sqrt{2} \, y \rightarrow y = \frac{\eta - \Delta_T}{\sqrt{2}} \\
 \rightarrow 2 \, \gamma(\eta - \Delta_T) &= 2 \, \gamma(\sqrt{2} \, y) = 2 \, P_D = 1 - \text{erf}(y) \\
 \rightarrow \text{erf}\left(\frac{\eta - \Delta_T}{\sqrt{2}}\right) &= 1 - 2 \, P_D \rightarrow \text{erfinv}(1 - 2 \, P_D) = \frac{\eta - \Delta_T}{\sqrt{2}} \\
 \rightarrow \Delta_T &= \eta - \sqrt{2} \, \text{erfinv}(1 - 2 \, P_D).
 \end{aligned} \tag{149}$$

Now we can summarize the binary hypothesis testing algorithm that uses this new test statistic.

Step 1 - First, we would initialize the test statistic and the discrimination measure using:

$$\begin{aligned}
 \mathcal{L}(r(t_0)) &= 0 \\
 \Delta(t_0) &= 0.
 \end{aligned} \tag{150}$$

Step 2 - Then, at each time interval we would compute the following:

$m_o(t_i)$  and  $m_I(t_i)$  defined in Eq (119), using Eq (40) and Eq (41),

$d(t_i)$  and  $L(r(t_i))$  using Eq (124) and the unnumbered equation above it,

$$D(t_i) = \mathbf{d}(t_i)^T \mathbf{A}_k^{-1} \mathbf{d}(t_i),$$

$$\mathcal{L}(\mathbf{r}(t_i)) = L(\mathbf{r}(t_i)) + \mathcal{L}(\mathbf{r}(t_{i-1})),$$

$$\text{and } \Delta(t_i) = D(t_i) + \Delta(t_{i-1}).$$

Step 3 - We would then compare  $\Delta(t_i)$  to  $\Delta_T$ , as follows:

Step 3a - If  $\Delta(t_i) < \Delta_T$ , then a hypothesis test is not performed for this time iteration.

Step 3b - If  $\Delta(t_i) \geq \Delta_T$ , then a hypothesis test is triggered (hence the name trigger

discrimination measure) by comparing  $\mathcal{L}(\mathbf{r}(t_i))$  to  $\eta$  as follows:

Step 3b - 1 - If  $\mathcal{L}(\mathbf{r}(t_i)) > \eta$ , choose  $h_l$  ( $h_l$  becomes the primary hypothesis) and reset

the test statistic and discrimination measure using:

$$\begin{aligned} \mathcal{L}(\mathbf{r}(t_i)) &= 0 \\ \Delta(t_i) &= 0. \end{aligned} \tag{151}$$

Step 3b - 2 - If  $\mathcal{L}(\mathbf{r}(t_i)) \leq \eta$ , choose  $h_o$  ( $h_o$  remains the primary hypothesis) and reset

the test statistic and discrimination measure using Eq (151).

This algorithm performs a binary hypothesis test, but it can be extended to any  $N$ -ary test by using the algorithm against several alternative hypotheses. We start by defining the mean of the residual given a specific alternative hypothesis as:

$$E_{Z(t_{i-1})} \{ \mathbf{r}(t_i) \} |_{\mathbf{h}_k} = \mathbf{m}_k(t_i) \tag{152}$$

where  $k$  denotes the specific alternative hypothesis. We will define  $\mathcal{L}_k(\mathbf{r}(t_i))$  as the test statistic and  $\Delta_k(t_i)$  as the discrimination measure for the  $k^{\text{th}}$  hypothesis. Note that all of the alternative hypothesis

tests are run in parallel, so all of the test statistics and discrimination measures are computed in parallel (not sequentially) with those of the other hypotheses.

Step 1 - We initialize these measures for each of the alternative hypotheses:

$$\left. \begin{aligned} \mathcal{L}_k(r(t_i)) &= 0 \\ \Delta_k(t_i) &= 0 \end{aligned} \right\} \forall k. \quad (153)$$

Step 2 - At each time interval we compute  $m_k(t_i)$ , defined in Eq (152), using Eq (40) and Eq (41), and then perform all of the following computations for all of the alternative hypotheses:

$$\left. \begin{aligned} d_k(t_i) &= m_k(t_i) - m_0(t_i) \\ L_k(r(t_i)) &= d_k(t_i)^T A_k^{-1} r(t_i) - \frac{1}{2} d_k(t_i)^T A_k^{-1} (m_k(t_i) + m_0(t_i)) \\ D_k(t_i) &= d_k(t_i)^T A_k^{-1} d_k(t_i) \\ \mathcal{L}_k(r(t_i)) &= L_k(r(t_i)) + \mathcal{L}_k(r(t_{i-1})) \\ \Delta_k(t_i) &= D_k(t_i) + \Delta_k(t_{i-1}) \end{aligned} \right\} \forall k. \quad (154)$$

Initially, we will include all of the alternative hypotheses in a list of viable primary hypotheses. We will now start to remove alternative hypotheses that are not viable at this particular data sample.

Step 3 - First, we remove the alternative hypotheses that cannot be discriminated from the primary hypothesis from the list of viable primary hypotheses. For each of the alternative hypotheses,  $k$ , we would compare  $\Delta_k(t_i)$  to  $\Delta_T$ , as follows:

Step 3a - If  $\Delta_k(t_i) < \Delta_T$ , then a viable hypothesis test cannot be performed at this time iteration for this alternative hypothesis, so we remove this alternative hypothesis from the list of viable primary hypotheses.

Step 3b - If  $\Delta_k(t_i) \geq \Delta_T$ , then a viable hypothesis test can be performed at this time iteration, so we keep the alternative hypothesis  $k$  on the list of viable primary hypotheses.

Step 4 - Now we perform a hypothesis test between the current primary hypothesis and all of the alternative hypotheses left on the viable primary hypothesis list. For each of the alternative hypotheses left on the list of viable primary hypotheses, we perform a hypothesis test by comparing  $\mathcal{L}_k(r(t_i))$  to  $\eta$  as follows:

Step 4a - If  $\mathcal{L}_k(r(t_i)) \leq \eta$ , the alternative hypothesis is removed from the list of viable primary hypotheses and its test statistic and discrimination measure are reset, only for this particular alternative hypothesis. Note that the alternative hypotheses that were removed from the list of viable primary hypotheses did not trigger a hypothesis test, so we do not reset their test statistic and discrimination measure.

Step 4b - If  $\mathcal{L}_k(r(t_i)) > \eta$ , keep the alternative hypothesis on the list of viable primary hypotheses.

Step 5 - We now have two possible cases, either the list of viable primary hypotheses still has one or more entries, or it does not.

Step 5a - If the list of viable primary hypotheses is empty, then the current primary hypothesis is still in effect.

Step 5b - If the list of viable primary hypotheses is not empty, then we choose the hypothesis that has the largest test statistic and reset all of the test statistics and discrimination measures using:

$$\left. \begin{array}{l} \mathcal{L}_k(r(t_i)) = 0 \\ \Delta_k(t_i) = 0 \end{array} \right\} \forall k. \quad (155)$$

Step 4a allows the other alternative hypotheses to trigger the hypothesis test at the appropriate number of iterations to provide the desired hypothesis testing performance. Thus, the hypotheses that have discrimination measures that grow faster because they are easily distinguished from the primary hypothesis would be tested before the less easily distinguished hypotheses.

For example, let the primary hypothesis be a no flight control failure and two alternative hypotheses a left and right aileron failure. If the right aileron failure hypothesis discrimination measure grew faster than the left aileron failure hypothesis discrimination, a no failure hypothesis versus a right aileron failure hypothesis test would be triggered before the no failure vs. left aileron failure hypothesis test. Once the test is triggered, if the no failure hypothesis is chosen, then the test discrimination measure for the right aileron failure hypothesis is reset. The test discrimination measure for the left aileron failure hypothesis is not reset so it will continue to grow until a no failure vs. left aileron failure hypothesis test is triggered. Note that if the left aileron failure hypothesis discrimination measure were reset in this case, a no failure vs. left aileron failure hypothesis test would never be triggered. If the no failure vs. right aileron failure hypothesis test is triggered and the alternative hypothesis (right aileron failure) is chosen, then all of the test measures and hypothesis discrimination measures are reset because we would have a new primary hypothesis against which we will test the alternative hypotheses.

Note that this is not truly, an optimal failure hypothesis testing structure because we are only testing the primary hypothesis against the alternative hypotheses, and we are not testing any alternative hypotheses against each other. In the example above, we are not testing the left vs. right aileron failures. This could artificially boost the likelihood of an incorrect hypothesis in the event of a missed detection. For example, if a no-failure vs. right aileron hypothesis test is triggered, and a

right aileron failure is mistakenly missed, then the left aileron failure discrimination measure is not reset and most likely will reach its discrimination trigger before the right aileron failure discrimination measure (which was reset after the test was conducted) reaches its trigger. If there happens to be ambiguity between the two hypotheses (right vs. left aileron failure), the right aileron vs. no-failure hypothesis test will likely choose the incorrect hypothesis (the right aileron failure). This occurs because we are not tracking the discrimination between the alternative hypotheses, which would require  $K(K-1)$  (where  $K$  is the number of hypotheses) different hypothesis triggers, discrimination measures, and tests. If it is known that certain alternate hypotheses have ambiguity problems, the probability of missed detection ( $1-P_D$ ) for those hypotheses could be set lower than the other hypotheses to decrease the probability of this type of problem.

The timing of the failure could influence the amount of time needed to identify the failure. Ideally, if the failure occurs right when the test discrimination measure is started, this algorithm will trigger a hypothesis test just at the time when the hypothesis is distinguishable from the primary hypothesis, say  $M$  time samples after the failure. However, if the failure occurred after the test discrimination measure has already started, the discrimination measure will trigger a test before the two hypotheses are truly distinguishable, so it is quite possible for the algorithm to miss the failure detection at this first test, but it should pick up the failure  $M$  time samples later. Thus, the worst case delay in the failure detection of this algorithm would be  $2M-1$ .

### 3.4 Chapter Summary

We have established the theoretical and algorithmic contributions of this research in this chapter. We started by presenting the Kalman filter equations and laid out the nomenclature that we use to represent the mismodeling in the models that are used to develop the Kalman filter equations. The specific mismodeling cases that we represented are a mismodeled input matrix, output matrix,

and state transition matrix. We developed expressions for the mean and covariance of the residual for any of these three types of mismodeling, and showed how the expressions reduced for specific cases of mismodeling. In Section 3.2.5 we developed an equivalent structure to the Standard Kalman Filter Bank (SKFB) that uses the residual and state estimates from a single Kalman filter, to compute the equivalent residuals of all other Kalman filters with known model differences, which would usually be implemented in an MMAE algorithm. We developed a Residual Correlation Kalman Filter Bank (RCKFB) in Section 3.2.6, that estimates the power spectral density of the residuals and feeds this to the hypothesis testing algorithm, motivated by the idea that this structure would exploit the known characteristics of the correlation of the residual to identify failures. Section 3.3.1 defined the Standard Hypothesis Testing Algorithm (SHTA) and described several of the modifications that have been researched. We developed a Neyman-Pearson based Hypothesis Testing Algorithm (NPHTA) in Section 3.3.2, where we extended the usual single time sample, binary hypothesis test formulation, to multiple time sample, multiple hypothesis testing. These developments suggest an algorithm architecture of an "MMAE-like" algorithm, but with the many elemental filters and residuals of a usual MMAE replaced by one filter, linear transformations (optimized "matched filters") and equivalent residuals. An eventually implemented system might be composed of an algorithm structure based on either the SKFB or RCKFB (or *both*), possibly using equivalent residuals rather than actual residuals, and perhaps using the NPHTA rather than the SHTA. The next chapter applies these various algorithms to a specific sensor/actuator failure detection problem to assess relative strengths of the various configurations.

## IV. Results

### 4.1 Chapter Overview

In this chapter we present the results of our research. In Section 4.2, we verify our development of an expression for the mean of the residual (Eq (40) and Eq (41) from Section 3.2.4) and the covariance of the residual (Eq (34) from Section 3.2.3). We present a graphical comparison of the time history of the predicted mean and covariance for each element of a residual along with a single sample of a residual from a Kalman filter. A more detailed comparison of the mean and covariance of the residual is made by temporally averaging five samples of the residual and then ensemble averaging these temporal averages, and ensemble averaging 10 samples of the residual and temporally averaging these ensemble (sample) averages. These empirically computed statistics are compared to the predicted values of the mean and covariance. This comparison is made for the specific cases of no mismodeling in the Kalman filter model, a mismodeled input control matrix (actuator failure), and a mismodeled output matrix (sensor failure). Time limitations prevented the computation of results for the mismodeled state transition matrix test case. The development of equivalent residuals, presented in Section 3.2.5, is verified in Section 4.3. We compare the residuals from a bank of several Kalman filters with the equivalent residuals that are computed using a single Kalman filter residual and known model differences between the Kalman filters. The comparison is made for the same specific cases of mismodeling that were presented in Section 4.2. These results suggest a new filter bank architecture that replaces the many filters and residuals with a single Kalman filter, several linear transforms, and equivalent residuals.

Section 4.4 is a comparison of the failure identification performance of the three different MMAE structures that were developed: Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA), Residual Correlation Kalman Filter Bank (RCKFB) with a SHTA, and SKFB with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA). Time limitations prevented the development and testing of a fourth configuration, namely a RCKFB with NPHTA. The testing of the failure identification performance was accomplished for various actuator failures, which were modeled by a column of the control input matrix being set to zero. Previous research [19, 31, 41] has shown that these were the most difficult failure modes to identify for this particular aircraft, so we chose to test the performance for these failures to be able to characterize the worst case performance of these various structures. The comparison of these different architectures suggests an algorithm structure that uses both the SKFB and RCKFB, perhaps with equivalent residuals instead of the actual residuals, using a NPHTA for the best failure identification performance.

## 4.2 Characterization of the Residual

In Section 3.3, we developed a recursive algorithm for estimating the mean and covariance of the Kalman filter residuals under various forms of hypothesized mismodeling. We present a comparison of these estimates with actual Kalman filter residuals. The comparisons are presented for three hypotheses: no mismodeling, an input control matrix mismodeling ( $\Delta B \neq 0$ ), and an output matrix mismodeling ( $\Delta H \neq 0$ ). For the flight control failure identification application, these hypotheses correspond to no flight control failure, an actuator failure, and a sensor failure, respectively.

The application that we used to test the theory developed in this dissertation consisted of a bank of 16 Kalman filters, each modeling a different hypothesis (1 no failure, 6 actuator failures, and

9 sensor failures). The complete set of results for this section would require us to present 256 test cases, 16 Kalman filter residuals for each of the 16 hypotheses. Therefore, we will present *representative* results to show the accuracy of the estimates of a Kalman filter residual for the three cases of mismodeling.

The plots presented below show each of the scalar elements of a single sample of the 9-element residual in a separate plot. The elements are shown as a function of time, with the simulation time being 8 seconds. The covariance of the residual is precomputed using Section 3.2.3. For each element of the residual, we superimpose on the plot of the single sample of the residual process, the computed mean, the computed mean plus the square root of the computed variance (corresponding to one computed standard deviation for the residual element), and the computed mean minus one standard deviation. Since the residual is assumed to be Gaussian, we would expect that about 68% of the time samples of the residual would lie within the plotted mean  $\pm$  one standard deviation bounds. Thus, we can visually compare the residual with its computed mean and covariance.

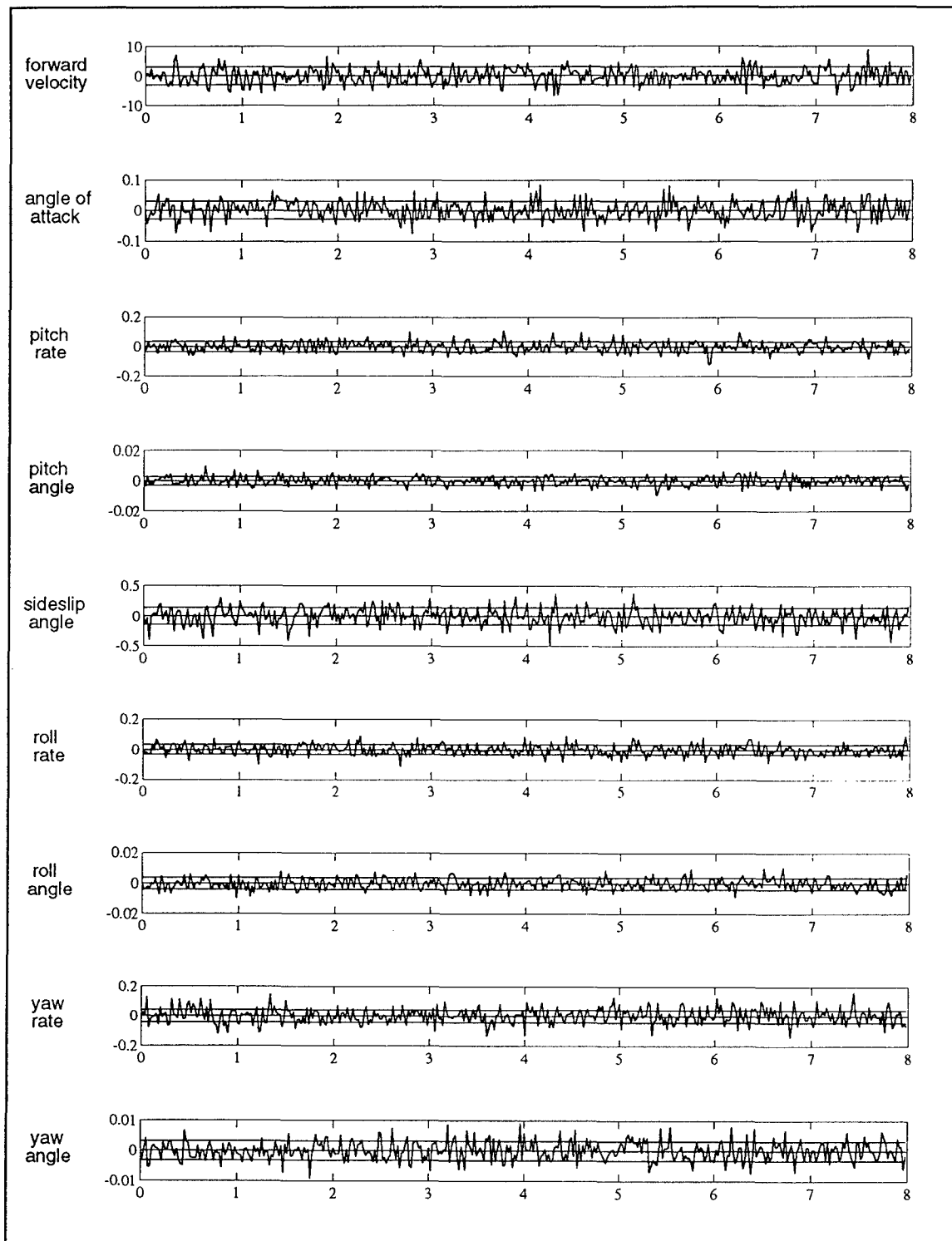
A more detailed analysis of the residual characteristics is accomplished by comparing various empirical statistics to the computed mean and covariance. First, we computed the temporal average (ergodic approximation of the mean) and the temporally-computed standard deviation of each of five realizations of the residual process. Then we averaged these temporal averages to get the ensemble-temporal average of the mean and standard deviation. Next, we computed the ensemble (sample) average of 10 realizations of the residual process, which gave us a time history average of both the mean and standard deviation. These ensemble (sample) averages were averaged temporally to obtain the temporal-ensemble averages. These two types of averages are compared to the predicted mean and standard deviation.

We chose to examine the residual vector from the Kalman filter that models a fully functional aircraft (no flight control failures). The true system model assumes a fully functional aircraft for the first second of simulation, followed by a specific single flight control failure. Thus, for the first second of simulation, the chosen Kalman filter has no mismodeling, after which a specific mismodeling occurs, which will be the difference between the fully functional aircraft model and the actual flight control failure model. We will look at specific cases of mismodeling in the following sections.

4.2.1 No Mismodeling (No Failure). We start by comparing the residual from the Kalman filter that models a fully functional aircraft (no flight control actuator or sensor failures) where the model is accurate throughout the simulation time. The mean and covariance were computed using Eq (40) and Eq (41) with  $\Delta B=0$ ,  $\Delta H=0$ , and  $\Delta \Phi=0$ . The results for this specific case of mismodeling were developed in Section 3.2.4.1, where we found that the residual should be zero-mean for all time. Figure 12 shows that the residual is zero-mean throughout the simulation and approximately 68% of the time samples are within the computed standard deviation bounds.

To facilitate a more accurate comparison of the residual with its computed characteristics, we present, in Table 1, the computed the mean and standard deviation of the scalar components, the ensemble-temporal averages, and the temporal-ensemble averages. The computed standard deviation of each element was calculated by computing the diagonal elements of the matrix square root of the computed covariance matrix (which was shown to be time invariant).

These results clearly indicate that the residual was well characterized for the case of no mismodeling in the Kalman filter model. Note that these averages (computed using both methods) may differ because of the different number samples used for each. Also, these averages are



**Figure 12.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with No Mismodeling

**Table 1**

Kalman Filter Residual Statistical Comparison when No Mismodeling Exists.

Residual Element	Computed Average	Ensemble Temporal Average	Temporal Ensemble Average	Computed $\sigma$	Ensemble Temporal $\sigma$	Temporal Ensemble $\sigma$
Forward Velocity	0	-1.34 E-1	-2.24 E-1	3.00	2.97	2.96
Angle of Attack	0	3.70 E-3	-8.04 E-4	2.96 E-2	2.95 E-2	2.95 E-2
Pitch Rate	0	4.09 E-4	4.65 E-3	3.55 E-2	3.53 E-2	3.50 E-2
Pitch Angle	0	5.76 E-4	3.80 E-5	2.92 E-3	2.87 E-3	2.97 E-3
Sideslip Angle	0	-3.64 E-3	-6.51 E-3	1.44 E-1	1.44 E-1	1.42 E-1
Roll Rate	0	-2.30 E-4	-1.75 E-3	3.46 E-2	3.41 E-2	3.48 E-2
Roll Angle	0	-1.62 E-5	-7.12 E-5	3.94 E-3	3.80 E-3	3.99 E-3
Yaw Rate	0	9.67 E-5	5.66 E-4	5.32 E-2	5.57 E-2	5.40 E-2
Yaw Angle	0	-5.81 E-5	-5.34 E-5	3.22 E-3	3.12 E-3	3.20 E-3

dominated significantly by the standard deviation values, as we would expect from the computed mean (i.e., zero) and standard deviations.

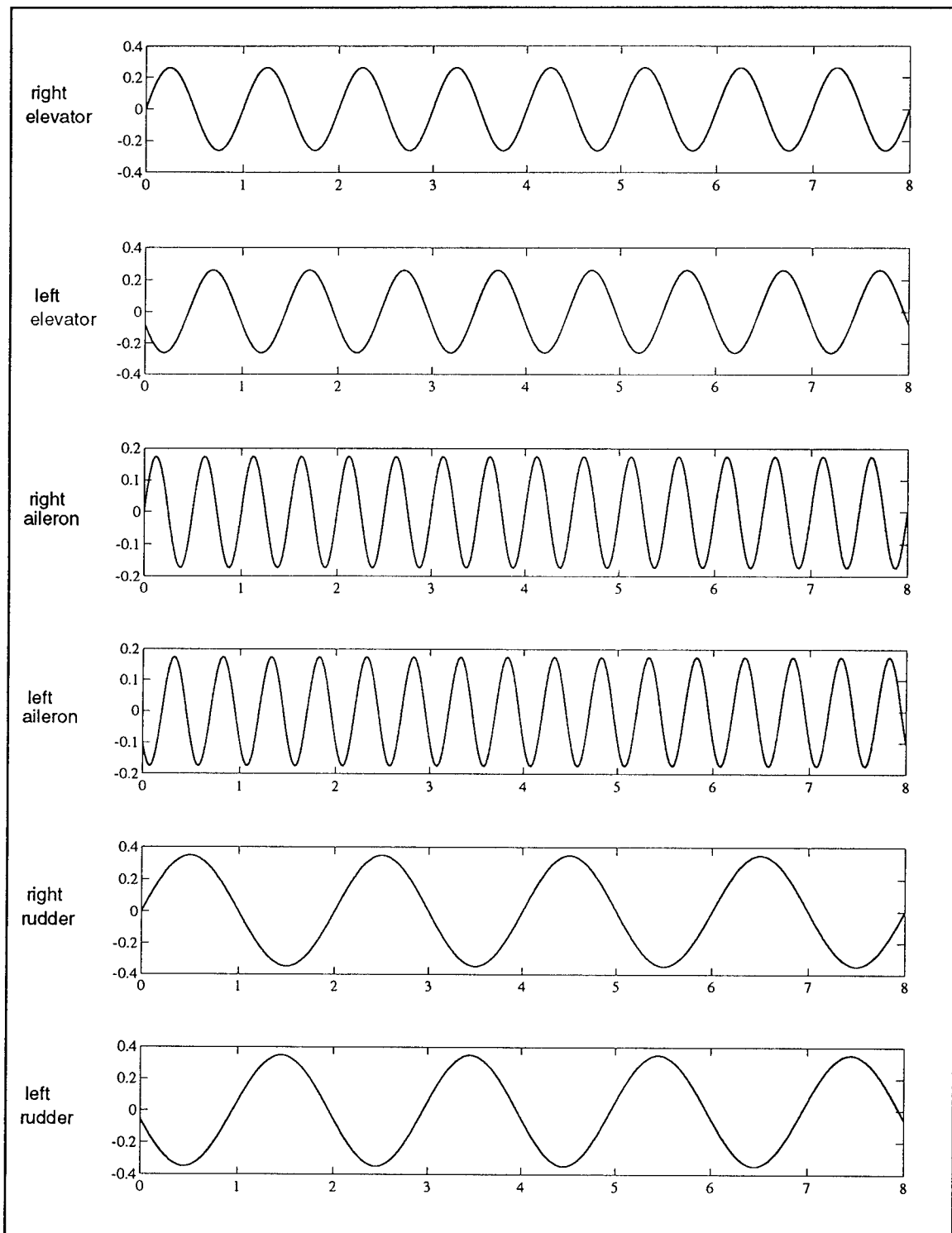
#### 4.2.2 Mismodeled Input Control Matrix (Actuator Failure). We now

compare the residual from the Kalman filter that uses a fully functional aircraft model to the computed mean and covariance assuming that the control input matrix is mismodeled starting at 1 second into the simulation. This situation corresponds to an actuator failure occurring at 1 second. We will look at an elevator actuator failure, an aileron actuator failure, and a rudder actuator failure. Similar results were obtained using the Kalman filter that incorrectly modeled the failure status from the start of simulation to 1 second, and then correctly modeled the failure status from then to the end of the simulation. An example of this case, where the filter model is based on a left elevator failure and the left elevator failure occurs at 1 second, is shown later in Figure 21, which shows that the

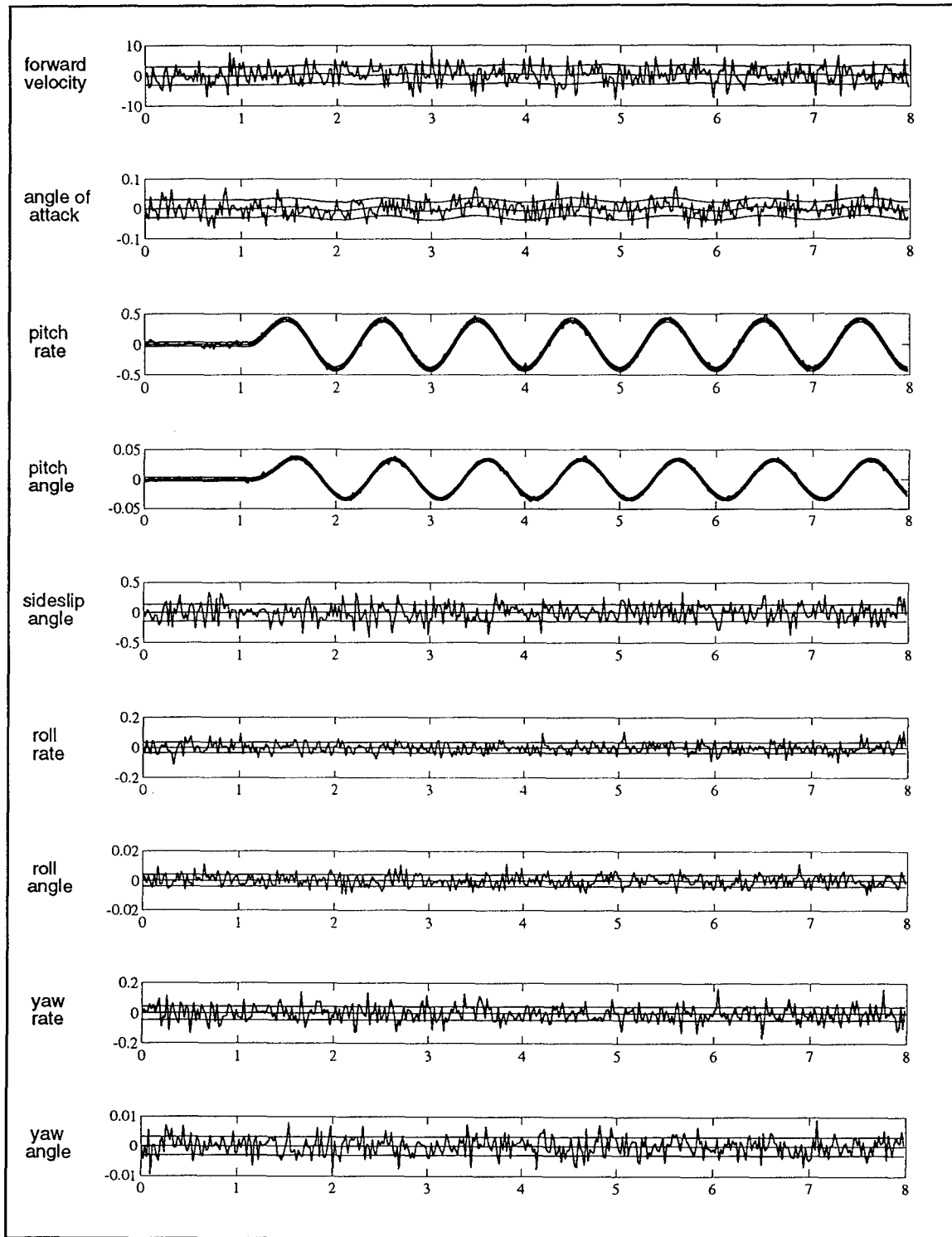
residual initially is not zero-mean, but eventually, after the failure occurs, becomes zero-mean. For all of these cases, the mean of the residual was computed using Eq (40) and Eq (41) and the expected covariance was developed in Section 3.2.3. In Section 3.2.4.2 we found that  $\Delta B \neq 0$  results in a residual mean that has components of the control input, including purposeful dithers that are put into the input to enhance failure identification. The inputs that were used for this research are shown in Figure 13.

The results for the elevator failure case, are graphically shown in Figure 14. For this case we have no mismodeling up to 1 second in the simulation, then mismodeling occurs in the column of the control input matrix that corresponds to an elevator, a left elevator in this case. The left elevator input is quite evident in the residual, particularly in the pitch rate and pitch angle where there is a significant change in scale. Also the forward velocity and angle of attack show a slight change in mean corresponding to the left elevator input. The other elements of the residual do not show the left elevator input because the aircraft model assumes there is no cross-coupling between the pitch axis and the other (lateral-directional) axes.

- These are precisely the results that we expected. If an elevator actuator failed, then the aircraft will not respond to the elevator input. The Kalman filter with the model based on a fully functional aircraft, would predict the measurements based on the aircraft responding to the elevator input. Since the aircraft is not responding to this input, the difference between the Kalman filter's prediction of the measurements and the actual measurements (the residual) would show the aircraft response. For this particular case, a sinusoidal elevator input would cause a change in the pitch rate, pitch angle, angle of attack, and forward velocity. Since the fully functional Kalman filter would predict these responses, but the aircraft is actually not responding to this elevator input, the difference between the filter's prediction and the actual measurements would show this expected



**Figure 13.** Control Inputs, Plotted in Radians.



**Figure 14.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Elevator Input.

response to the elevator input. This is precisely what we found in Figure 14. Similar results were obtained for a right elevator failure.

To facilitate further analysis, we computed the bias-compensated residual by differencing the computed mean and the value of the realization of the residual process for each time sample:

$$\xi(t_i) = m_c(t_i) - r(t_i) \quad (156)$$

where  $m_c(t_i)$  is the computed residual mean. If the computed residual mean accurately represents the residual mean, then we would expect that the ensemble mean of the temporal average of this difference to be:

$$\begin{aligned} E_{Z(t_{i-1})} \left\{ \frac{1}{M} \sum_{j=t-M+1}^t \xi(t_j) \right\} &= E_{Z(t_{i-1})} \left\{ \frac{1}{M} \sum_{j=t-M+1}^t [m_c(t_j) - r(t_j)] \right\} = \frac{1}{M} \sum_{j=t-M+1}^t [m_c(t_j) - E_{Z(t_{i-1})} \{ r(t_j) \}] \\ &= \frac{1}{M} \sum_{j=t-M+1}^t [m_c(t_j) - m_c(t_j)] \\ &= 0 \end{aligned} \quad (157)$$

Likewise, the temporal average of the ensemble mean of this difference is:

$$\begin{aligned} \frac{1}{M} \sum_{i=1}^M E_{Z(t_{i-1})} \{ m_c(t_i) - r(t_i) \} &= \frac{1}{M} \sum_{i=1}^M [m_c(t_i) - E_{Z(t_{i-1})} \{ r(t_i) \}] \\ &= \frac{1}{M} \sum_{i=1}^M [m_c(t_i) - m_c(t_i)] \\ &= 0 \end{aligned} \quad (158)$$

Since the only random part of  $\xi$  is the residual, the covariance of this bias-compensated residual is the covariance of the residual.

Thus, if we compute the temporal-ensemble average (compute the ensemble average first and then compute the temporal average of the ensemble average) and the ensemble-temporal average (compute the temporal average of each realization and then compute the ensemble average of these temporal averages) we would expect the same results that were predicted for No Mismodeling case, shown in Table 1.

We computed the results for three cases, an elevator failure, an aileron failure, and a rudder failure. The results for the first case, an elevator failure, are shown in Table 2. Clearly, the residual is accurately characterized.

**Table 2**

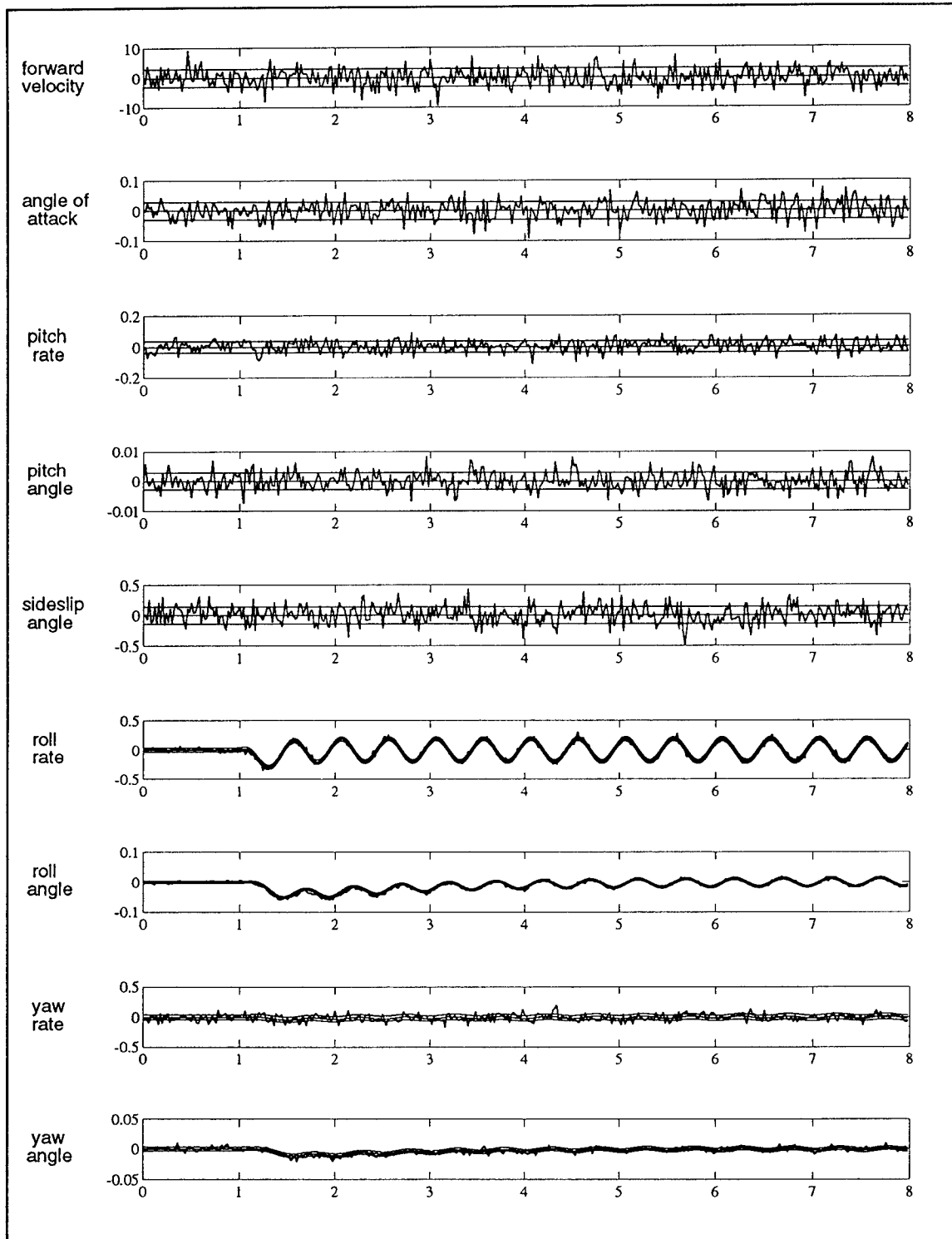
Kalman Filter Residual Statistical Comparison for the Case of an Elevator Failure.

Residual Element	Computed Average of $\xi$	Ensemble Temporal Average of $\xi$	Temporal Ensemble Average of $\xi$	Computed $\sigma$ of $\xi$	Ensemble Temporal $\sigma$ of $\xi$	Temporal Ensemble $\sigma$ of $\xi$
Forward Velocity	0	-2.05 E-1	-2.35 E-1	3.00	3.00	2.98
Angle of Attack	0	-4.83 E-3	-1.99 E-4	2.96 E-2	2.92 E-2	2.96 E-2
Pitch Rate	0	-3.70 E-3	-1.30 E-3	3.55 E-2	3.54 E-2	3.60 E-2
Pitch Angle	0	5.76 E-4	-9.60 E-5	2.92 E-3	2.76 E-3	2.87 E-3
Sideslip Angle	0	3.94 E-3	2.45 E-4	1.44 E-1	1.43 E-1	1.42 E-1
Roll Rate	0	-2.30 E-4	-4.03 E-4	3.46 E-2	3.38 E-2	3.44 E-2
Roll Angle	0	3.55 E-4	-7.78 E-5	3.94 E-3	3.87 E-3	3.97 E-3
Yaw Rate	0	3.22 E-5	5.66 E-4	5.32 E-2	5.49 E-2	5.43 E-2
Yaw Angle	0	-5.81 E-5	8.08 E-6	3.22 E-3	3.15 E-3	3.18 E-3

The next case corresponds to an aileron failure. Again, we have no mismodeling up to 1 second in the simulation, then mismodeling occurs in the column of the control input matrix that corresponds to an aileron. In Figure 15 we graphically show the results; the input to the left aileron is clearly evident in the residual, particularly in the roll rate and roll angle, and to a lesser degree in the yaw rate and yaw angle. Note the dramatic change in scale for these elements of the residual vector. Just like in the elevator failure case, the aircraft is not responding to the left aileron input, but the fully functional Kalman filter is predicting that the aircraft will respond to this input. Thus, the difference between the actual measurements (which show the aircraft's lack of response) and the predicted measurements (which assume that the aircraft is responding) will reflect the aircraft response to a left aileron input. The expected aircraft response to a sinusoidal left aileron input would be a sinusoidal change in the roll rate and roll angle. Also, since this is an input to only one aileron, we expect a small sinusoidal change in the yaw rate and yaw angle. This is precisely what was observed in Figure 15. Similar results were obtained for a right aileron failure.

We also computed and tabulated the ensemble-temporal average and temporal-ensemble average of the bias-compensated residual. These results, shown in Table 3, show that the characteristics of the residual are accurately computed for the case of an aileron failure.

The last case corresponds to a rudder failure. Again, we have no mismodeling up to 1 second in the simulation, then mismodeling occurs in the column of the control input matrix that corresponds to a rudder. In Figure 16 we show the results for a left rudder failure. The input to the left rudder is clearly evident in the residual, particularly in the yaw rate and yaw angle with some cross-coupling to the roll rate and roll angle. Again, note the change in scale for these residualelements. This aircraft has two rudders (similar to the A-10 aircraft), so if one rudder is not responding to a sinusoidal input, the other rudder would cause not only sinusoidal changes in yaw



**Figure 15.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Aileron Input.

**Table 3**

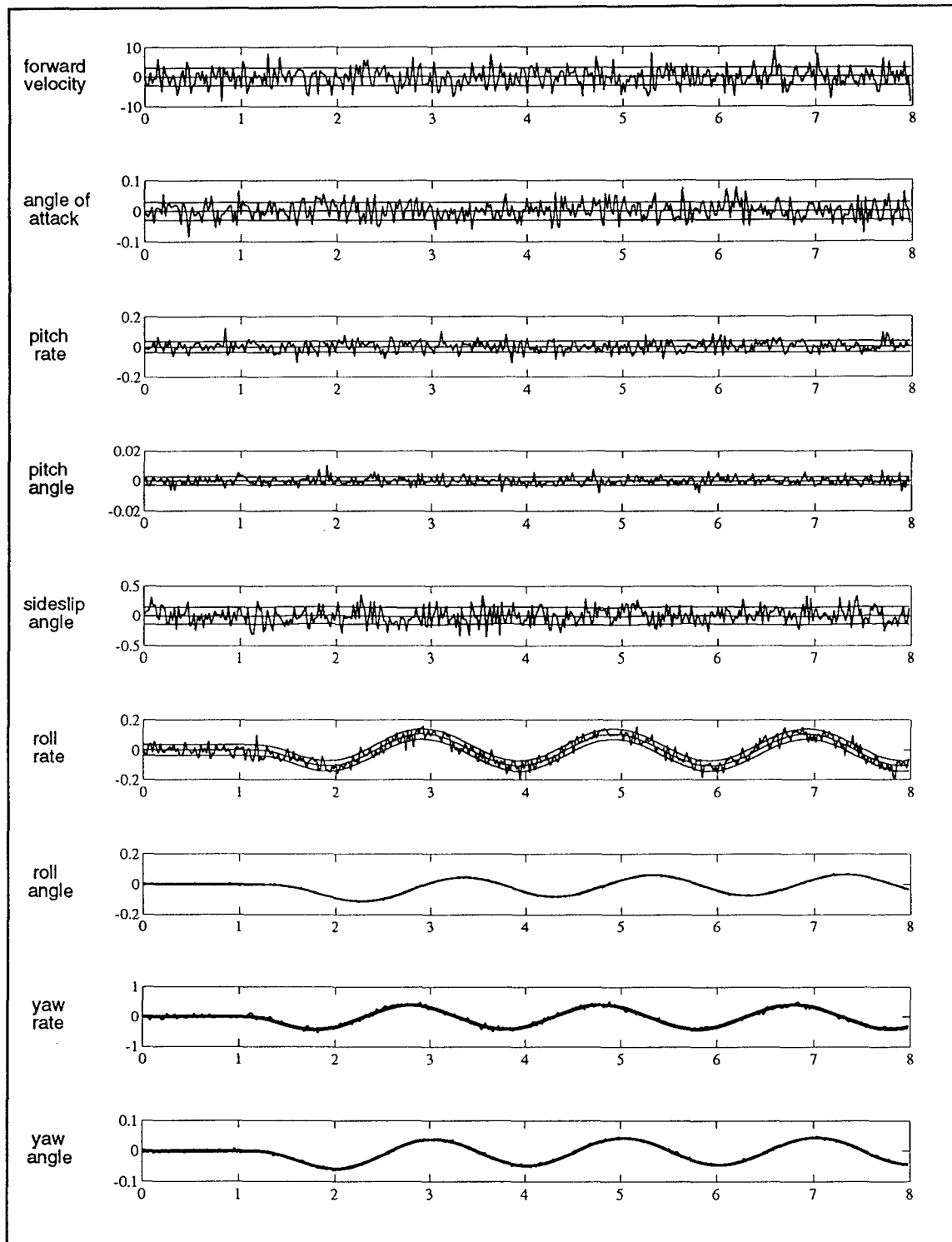
Kalman Filter Residual Statistical Comparison for the Case of an Aileron Failure.

Residual Element	Computed Average of $\xi$	Ensemble Temporal Average of $\xi$	Temporal Ensemble Average of $\xi$	Computed $\sigma$ of $\xi$	Ensemble Temporal $\sigma$ of $\xi$	Temporal Ensemble $\sigma$ of $\xi$
Forward Velocity	0	2.24 E-1	-9.50 E-2	3.00	2.96	2.96
Angle of Attack	0	-1.76 E-3	-1.08 E-4	2.96 E-2	2.91 E-2	2.78 E-2
Pitch Rate	0	1.38 E-3	6.54 E-4	3.55 E-2	3.58 E-2	3.44 E-2
Pitch Angle	0	4.68 E-5	3.23 E-4	2.92 E-3	2.85 E-3	2.89 E-3
Sideslip Angle	0	1.59 E-2	1.46 E-3	1.44 E-1	1.42 E-1	1.42 E-1
Roll Rate	0	1.10 E-3	9.89 E-4	3.46 E-2	3.45 E-2	3.48 E-2
Roll Angle	0	-2.76 E-4	2.05 E-3	3.94 E-3	3.67 E-3	3.75 E-3
Yaw Rate	0	1.13 E-3	2.86 E-3	5.32 E-2	5.52 E-2	5.46 E-2
Yaw Angle	0	3.18 E-3	6.09 E-4	3.22 E-3	3.19 E-3	3.19 E-3

**Table 4**

Kalman Filter Residual Statistical Comparison for the Case of a Rudder Failure.

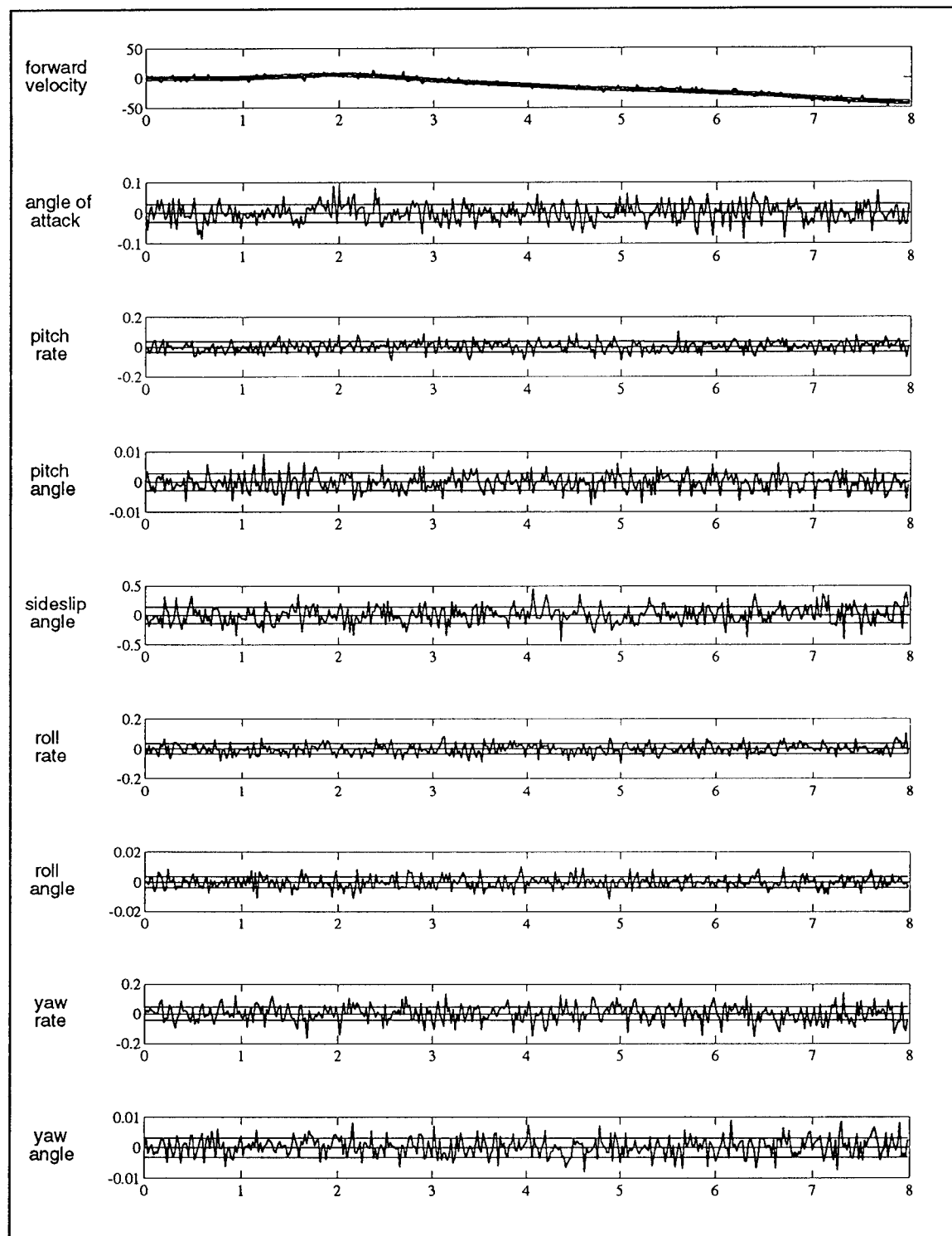
Residual Element	Computed Average of $\xi$	Ensemble Temporal Average of $\xi$	Temporal Ensemble Average of $\xi$	Computed $\sigma$ of $\xi$	Ensemble Temporal $\sigma$ of $\xi$	Temporal Ensemble $\sigma$ of $\xi$
Forward Velocity	0	-2.01 E-1	-1.85 E-1	3.00	2.95	2.98
Angle of Attack	0	-1.12 E-3	5.60 E-4	2.96 E-2	2.97 E-2	2.98 E-2
Pitch Rate	0	2.61 E-3	2.66 E-3	3.55 E-2	3.59 E-2	3.53 E-2
Pitch Angle	0	-7.83 E-5	4.19 E-5	2.92 E-3	2.94 E-3	2.86 E-3
Sideslip Angle	0	-2.16 E-3	-3.79 E-4	1.44 E-1	1.43 E-1	1.43 E-1
Roll Rate	0	6.07 E-4	6.09 E-4	3.46 E-2	3.53 E-2	3.48 E-2
Roll Angle	0	1.20 E-3	3.16 E-4	3.94 E-3	3.51 E-3	3.89 E-3
Yaw Rate	0	3.01 E-3	2.09 E-3	5.32 E-2	5.35 E-2	5.28 E-2
Yaw Angle	0	5.71 E-3	1.95 E-4	3.22 E-3	3.15 E-3	3.27 E-3



**Figure 16.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Rudder Input.

rate and yaw angle, but some small cross-coupling would occur in roll rate and roll angle. Again, since the fully functional Kalman filter predicts that the aircraft is responding to the input, but the aircraft actually is not, the residual would reflect the aircraft response to a single rudder input. This is precisely what is reflected in Figure 16. As before, the difference results were computed and are tabulated in Table 4. Clearly, the residual is accurately characterized. Similar results were obtained for a right rudder failure.

4.2.3 Mismodeled Output Matrix (Sensor Failure). We now compare the residual from the Kalman filter that uses a fully functional aircraft model to the computed mean and covariance, assuming that the output matrix is mismodeled starting at 1 second into the simulation. This situation corresponds to a sensor failure occurring at 1 second. We look separately at a failure in the forward velocity sensor and the pitch rate sensor. Once again, the mean of the residual was computed using Eq (40) and Eq (41) and the expected covariance was developed in Section 3.2.3. The failure mode that is represented by this mismodeling in the output matrix (a row of the matrix is zero) corresponds to a "fail to nominal," rather than "fail to zero," since the Kalman filter model is a perturbation model. The mismodeling in the output matrix zeros out any non-zero value of the perturbation state, so any perturbations from the nominal are set to zero for this failure mode. The results for a forward velocity sensor failure are graphically shown in Figure 17, where the forward velocity residual element is clearly no longer zero-mean (note the dramatic change in scale). The aircraft model did not include a thrust input, so we consistently noticed that the forward velocity decreased throughout the simulation (since there was no thrust to counteract the drag caused by the actuator inputs). Once the forward velocity sensor failed, this decrease in velocity was not measured, but the fully functional Kalman filter continued to predict that the forward velocity would decrease. Therefore, the difference between the actual measurement and the filter's prediction of that



**Figure 17.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Forward Velocity Sensor.

**Table 5**

Kalman Filter Residual Statistical Comparison for the Case of a Forward Velocity Sensor Failure.

Residual Element	Computed Average of $\xi$	Ensemble Temporal Average of $\xi$	Temporal Ensemble Average of $\xi$	Computed $\sigma$ of $\xi$	Ensemble Temporal $\sigma$ of $\xi$	Temporal Ensemble $\sigma$ of $\xi$
Forward Velocity	0	-2.01 E-1	2.27 E-1	3.003	2.98	3.05
Angle of Attack	0	-3.79 E-4	-1.01 E-3	2.96 E-2	2.89 E-2	2.97 E-2
Pitch Rate	0	8.55 E-4	9.41 E-2	3.55 E-2	3.47 E-2	3.66 E-2
Pitch Angle	0	-1.01 E-5	-6.13 E-3	2.92 E-3	2.87 E-3	2.94 E-3
Sideslip Angle	0	6.82 E-4	1.64 E-3	1.44 E-1	1.43 E-1	1.41 E-1
Roll Rate	0	1.55 E-4	4.86 E-4	3.46 E-2	3.41 E-2	3.38 E-2
Roll Angle	0	2.62 E-4	1.76 E-4	3.94 E-3	3.86 E-3	3.97 E-3
Yaw Rate	0	-9.14 E-4	-1.73 E-3	5.32 E-2	5.36 E-2	5.29 E-2
Yaw Angle	0	2.87 E-5	1.12 E-4	3.22 E-3	3.20 E-3	3.18 E-3

measurement is clearly reflected in the residual. The difference between the computed mean and the actual residual was computed and the temporal mean and standard deviation were calculated. The results are tabulated in Table 5. Once again, the characteristics of the residual are accurately computed.

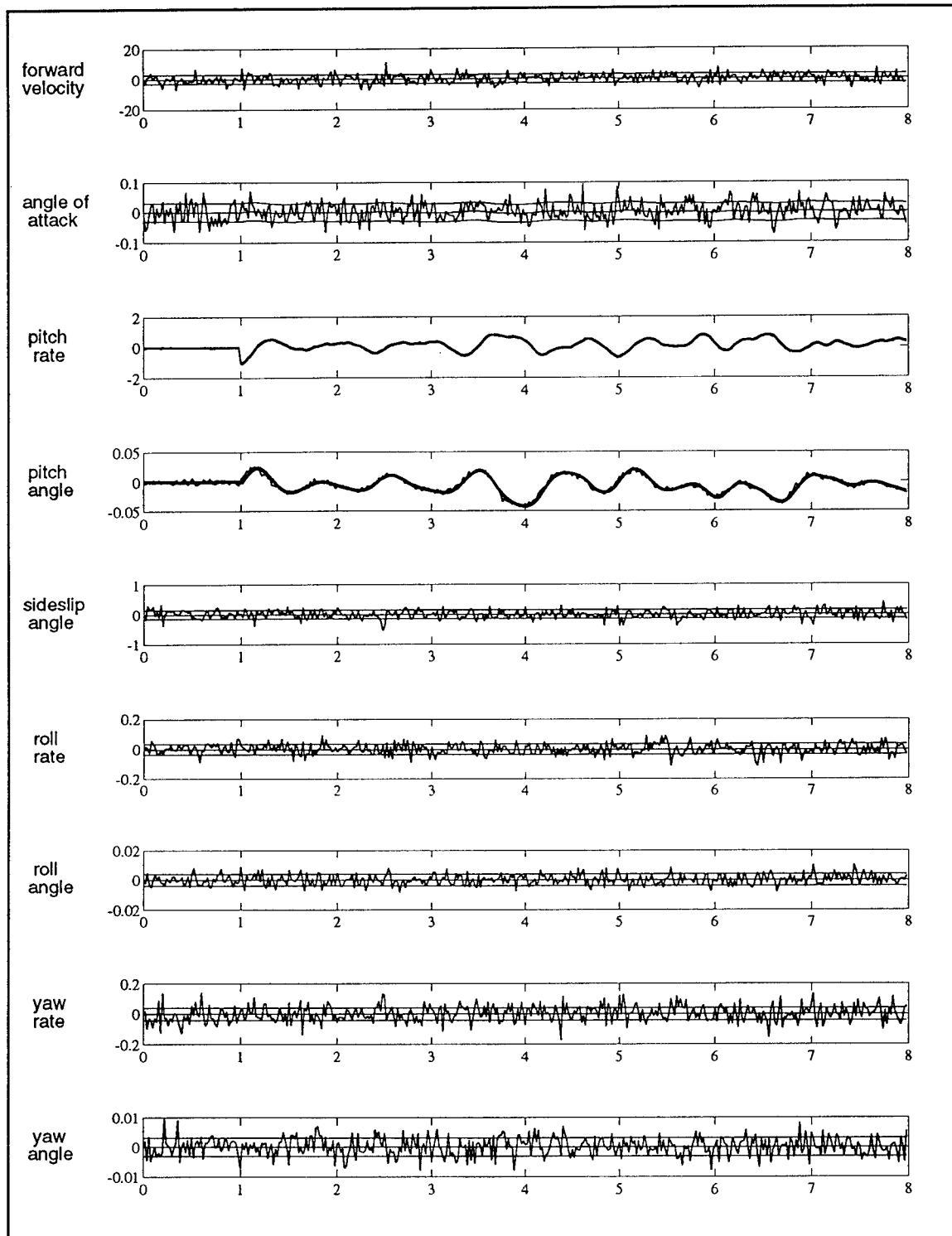
The results for the forward velocity sensor failure are representative only for the sensors that have very weak cross-coupling with other sensors: specifically, the angle of attack and sideslip angle sensors. The other sensors have stronger cross-coupling. This cross-coupling is not through the output matrix, but through the state transition matrix (i.e., the system dynamics and kinematics) and Kalman filter gain matrix. If the Kalman filter assumes that the pitch rate measurement is accurate (which the fully functional Kalman filter does), the state estimates are updated using this

**Table 6**

Kalman Filter Residual Statistical Comparison for the Case of a Pitch Rate Sensor Failure.

Residual Element	Computed Average of $\xi$	Ensemble Temporal Average of $\xi$	Temporal Ensemble Average of $\xi$	Computed $\sigma$ of $\xi$	Ensemble Temporal $\sigma$ of $\xi$	Temporal Ensemble $\sigma$ of $\xi$
Forward Velocity	0	7.25 E-2	2.29 E-1	3.003	2.98	2.97
Angle of Attack	0	3.12 E-4	-6.13 E-4	2.96 E-2	2.95 E-2	2.92 E-2
Pitch Rate	0	1.56 E-2	1.04 E-2	3.46 E-2	3.47 E-2	3.39 E-2
Pitch Angle	0	-1.08 E-3	-6.76 E-3	2.92 E-3	2.94 E-3	2.94 E-3
Sideslip Angle	0	7.02 E-4	-3.04 E-4	1.44 E-1	1.43 E-1	1.43 E-1
Roll Rate	0	2.67 E-4	3.57 E-4	3.46 E-2	3.43 E-2	3.46 E-2
Roll Angle	0	-3.49 E-5	-2.79 E-4	3.94 E-3	3.96 E-3	3.87 E-3
Yaw Rate	0	1.37 E-4	-4.86 E-3	5.32 E-2	5.38 E-2	5.31 E-2
Yaw Angle	0	1.20 E-5	-4.48 E-5	3.22 E-3	3.20 E-3	3.16 E-3

measurement, even if they are wrong! Thus, the other states are corrupted by the incorrect measurement of this single element of the measurement vector. The residuals not only reflect the difference between the actual measurement and the failed measurement, but also differences will occur between the coupled measurements and the filter prediction of those measurements. This is clearly evident in Figure 18, where the pitch angle element of the residual is impacted as well as the pitch rate element. The average and standard deviation of the difference between the computed residual mean and the actual residual are tabulated in Table 6.

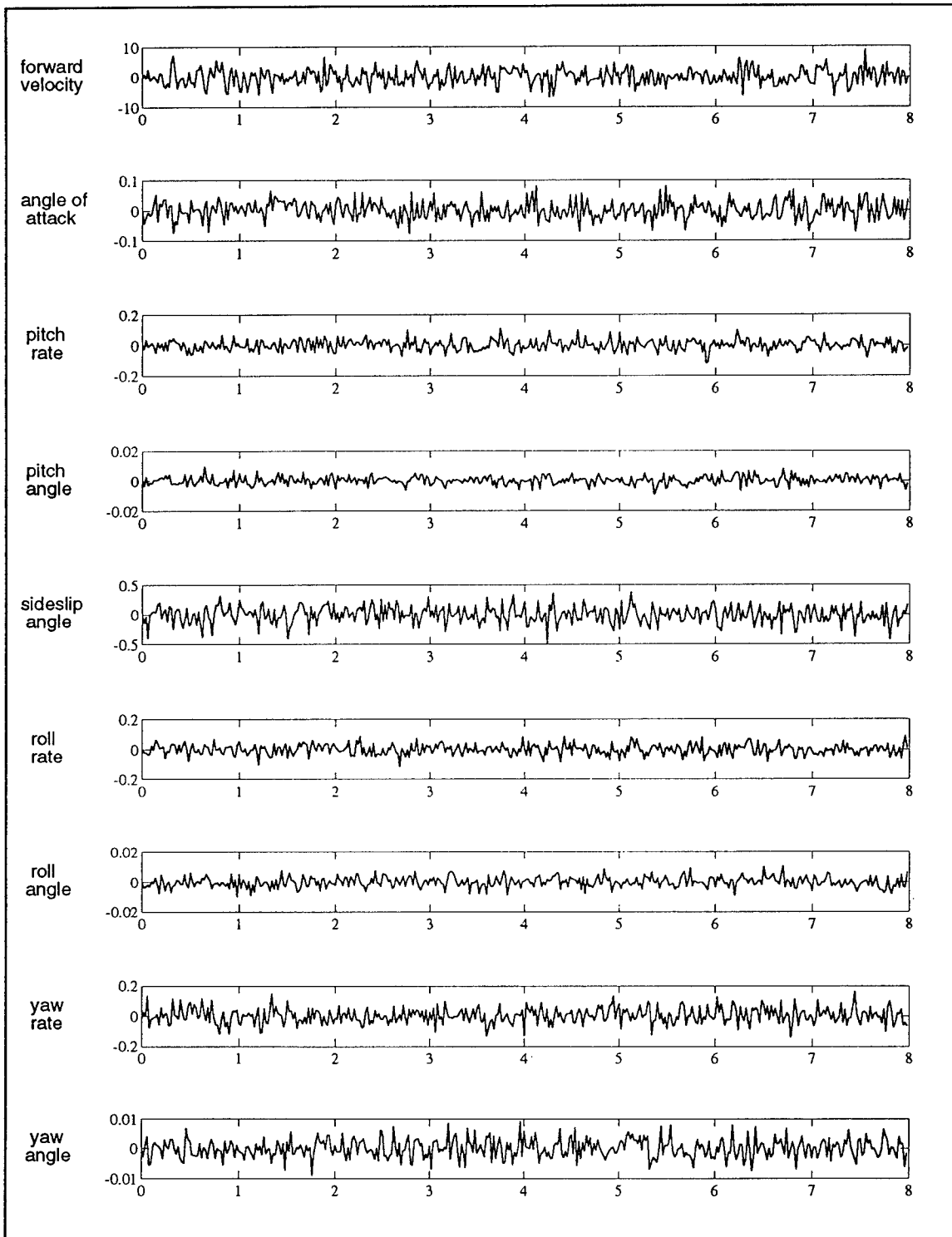


**Figure 18.** Residual and Computed Residual Mean and Standard Deviation of the Fully Functional Kalman Filter with a Mismodeled Pitch Rate Sensor.

### 4.3 Single Residual Kalman Filter Bank

In Section 3.2.5 we developed the framework for an MMAE structure that uses the residual vector from a single Kalman filter with a linear transform to produce the equivalent residual from another Kalman filter. In this section we present a comparison of the residual from a fully implemented Kalman filter with the computed equivalent of that residual. The application that we studied had an MMAE with 16 Kalman filters. Extensive simulation was conducted, since each of the 16 filters could be used to compute the residuals from any of the other 16 filters for any of the 16 possible failure scenarios. This results in 4096 test cases ( $16^3$ ), so we have elected to present a *representative* sample of three test cases.

4.3.1 No Mismodeling (No Failure). First, we present the test case in which the true system has no flight control failure and we compare the actual residual from the Kalman filter that assumed a fully functional aircraft with its equivalent residual computed using the residual from the Kalman filter that assumed a left elevator failure. This is a somewhat arbitrary choice to demonstrate the equivalency between the usual filter bank structure of several parallel Kalman filters, and a filter "bank" structure that uses a single Kalman filter with several linear transforms to produce equivalent residuals. We expect that the usual implementation of this equivalent structure would consist of the fully functional model based Kalman filter with the equivalent residuals computed for the Kalman filters that assume the various failures. Figure 19 shows what appears to be a set of single plots of the fully functional Kalman filter residual elements, but these plots are actually the equivalent residual plotted over the actual residual. It appears to be a single plot because the two residuals are equivalent. We computed the difference between the actual residual and the equivalent residual, and temporally averaged this difference. The results are tabulated in Table 7. Note that for some of the residual elements, the temporal average of the difference between the actual residual and



**Figure 19.** The Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Kalman Filter Residual for a No-Failure Test Case.

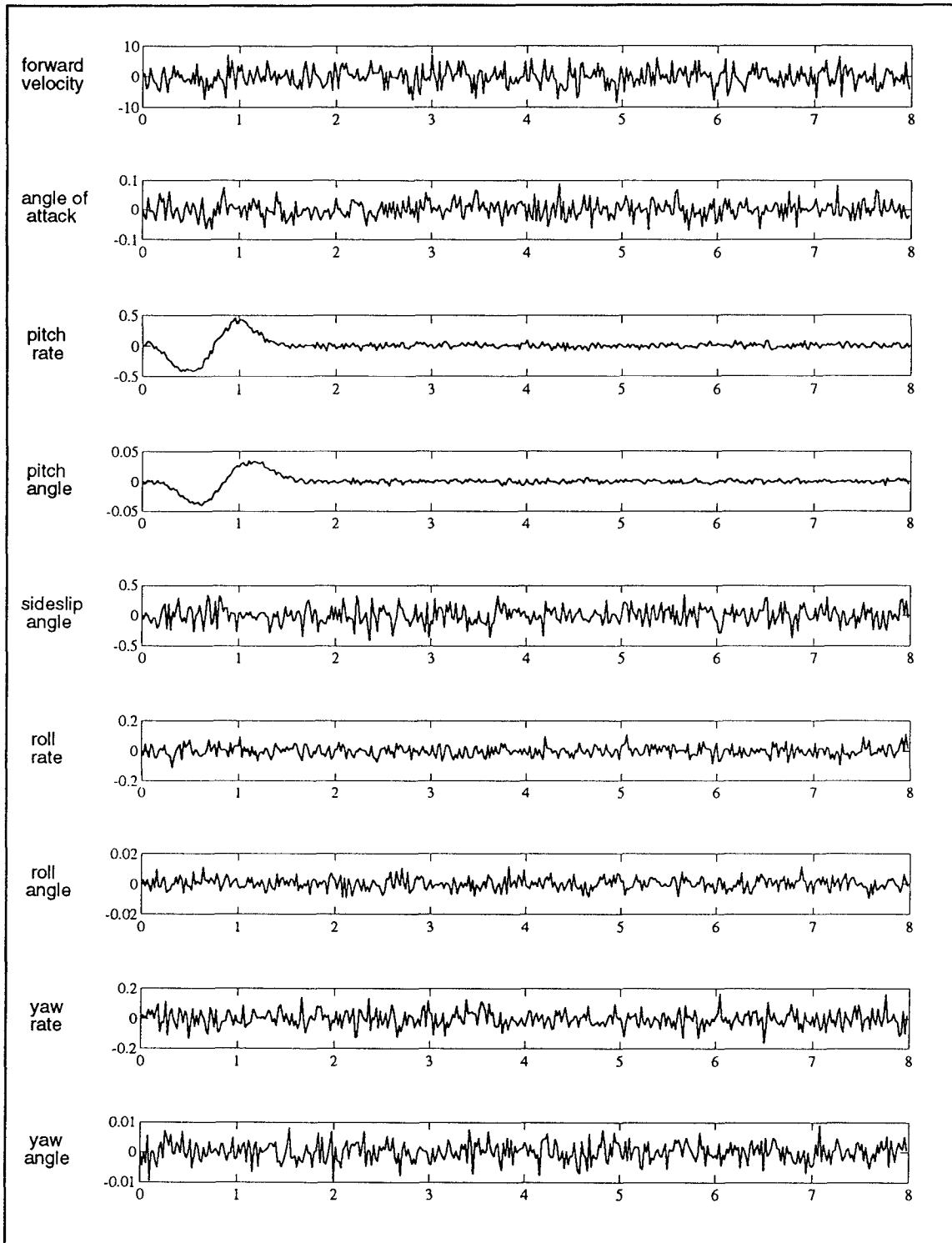
**Table 7**

Temporal Average of the Difference Between the Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Filter Residual for a No-Failure Test Case.

Residual Element	Temporal Average of Difference
Forward Velocity	-1.376 E-15
Angle of Attack	4.342 E-19
Pitch Rate	3.492 E-17
Pitch Angle	-5.926 E-18
Sideslip Angle	$< 10^{-21}$
Roll Rate	$< 10^{-21}$
Roll Angle	$< 10^{-21}$
Yaw Rate	$< 10^{-21}$
Yaw Angle	$< 10^{-21}$

equivalent residual is less than the precision of the simulation software ( $10^{-21}$ ), and in all cases the difference is many orders of magnitude less than the two individual residuals. Obviously, the actual residual and the computed equivalent residual are equivalent.

In Section 3.2.5, we proposed using these equivalent residuals to construct the Standard Kalman Filter Bank, where we are using the fully functional model based Kalman filter as the source filter for this implementation. We verified this construct in Figure 20 where we show the residual from the elevator failure Kalman filter along with its equivalent residual that was computed using the fully functional Kalman filter. This test case matches the structure shown in Figure 6 of Section 3.2.5, for the case of an elevator failure. The tabulated differences between the actual residual and



**Figure 20.** The Left Elevator Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case.

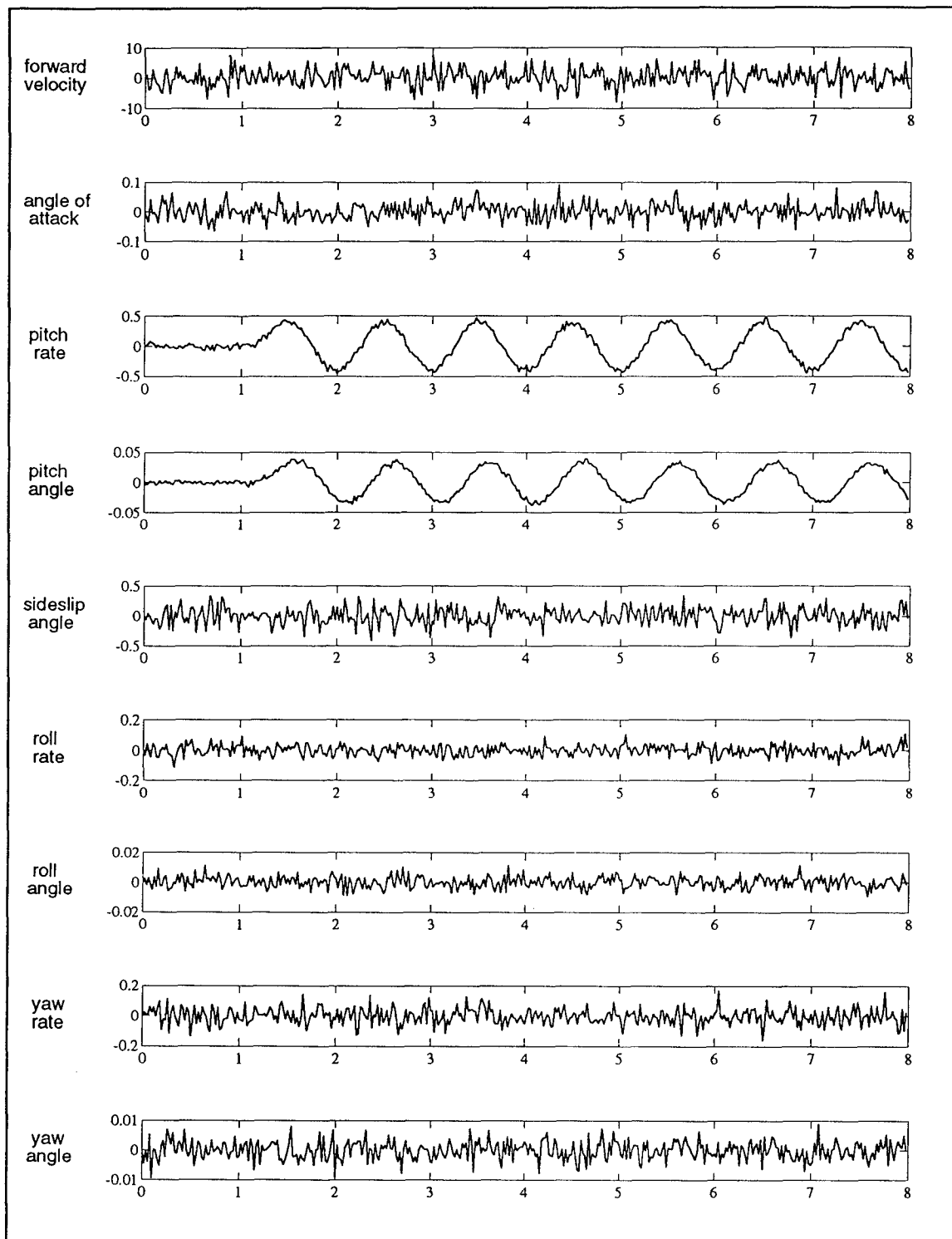
**Table 8**

Temporal Average of the Difference Between the Left Elevator Failure Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case.

Residual Element	Temporal Average of Difference
Forward Velocity	1.372 E-15
Angle of Attack	-4.516 E-19
Pitch Rate	-3.503 E-17
Pitch Angle	5.912 E-18
Sideslip Angle	$< 10^{-21}$
Roll Rate	$< 10^{-21}$
Roll Angle	$< 10^{-21}$
Yaw Rate	$< 10^{-21}$
Yaw Angle	$< 10^{-21}$

the equivalent residual are in Table 8; again, the equivalence of the two forms of the residual is evident.

4.3.2 Mismodeled Control Input Matrix (Actuator Failure). Next, we present the test case in which the true system has an actuator failure, specifically the left elevator, and we compare the actual residual from the Kalman filter that uses a fully functional aircraft model, with its equivalent residual computed using the residual from the Kalman filter that assumed a left elevator failure. Figure 21 again shows what appears to be a set of single plots of the fully functional Kalman filter residual, but these plots are actually the equivalent residual plotted over the actual



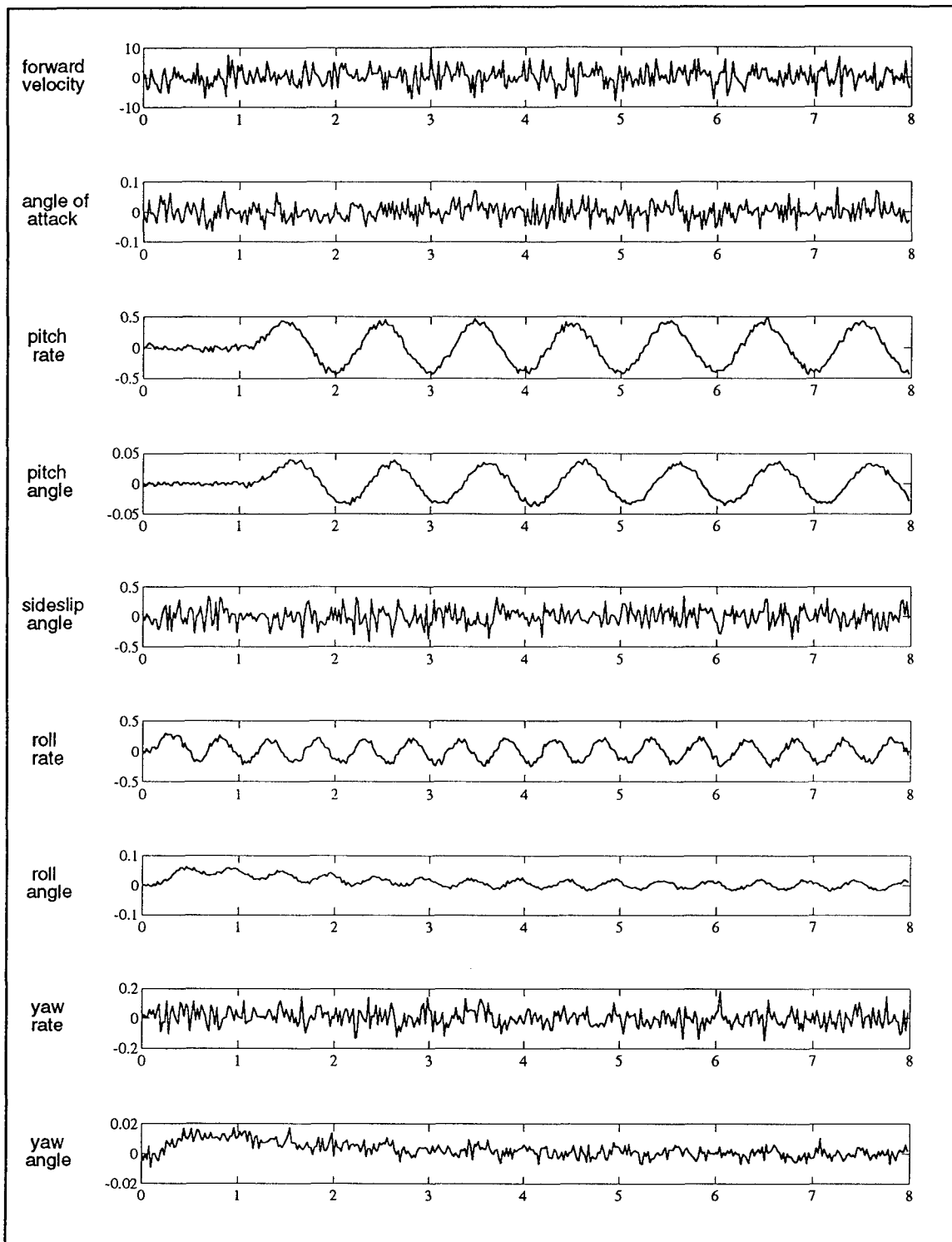
**Figure 21.** The Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Kalman Filter Residual for a Left Elevator Failure Test Case.

residual. We also tabulated the difference between the actual residual and the equivalent residual in Table 9. Once again, these residuals are equivalent.

To verify the structure developed in Section 3.2.5, we show the actual residual from the Kalman filter that assumes a left aileron failure when the actual failure is a left elevator, along with its equivalent residual computed using the fully functional Kalman filter. This comparison is made in Figure 22, and the differences are tabulated in Table 10. Clearly, the equivalent residual and the actual residual are identical.

**Table 9**  
Temporal Average of the Difference Between the Fully Functional Kalman Filter Residual and the Equivalent Residual Computed Using the Left Elevator Failure Kalman Filter Residual for a Left Elevator Failure Test Case.

Residual Element	Temporal Average of Difference
Forward Velocity	1.372 E-15
Angle of Attack	-4.516 E-19
Pitch Rate	-3.503 E-17
Pitch Angle	5.912 E-18
Sideslip Angle	$< 10^{-21}$
Roll Rate	$< 10^{-21}$
Roll Angle	$< 10^{-21}$
Yaw Rate	$< 10^{-21}$
Yaw Angle	$< 10^{-21}$



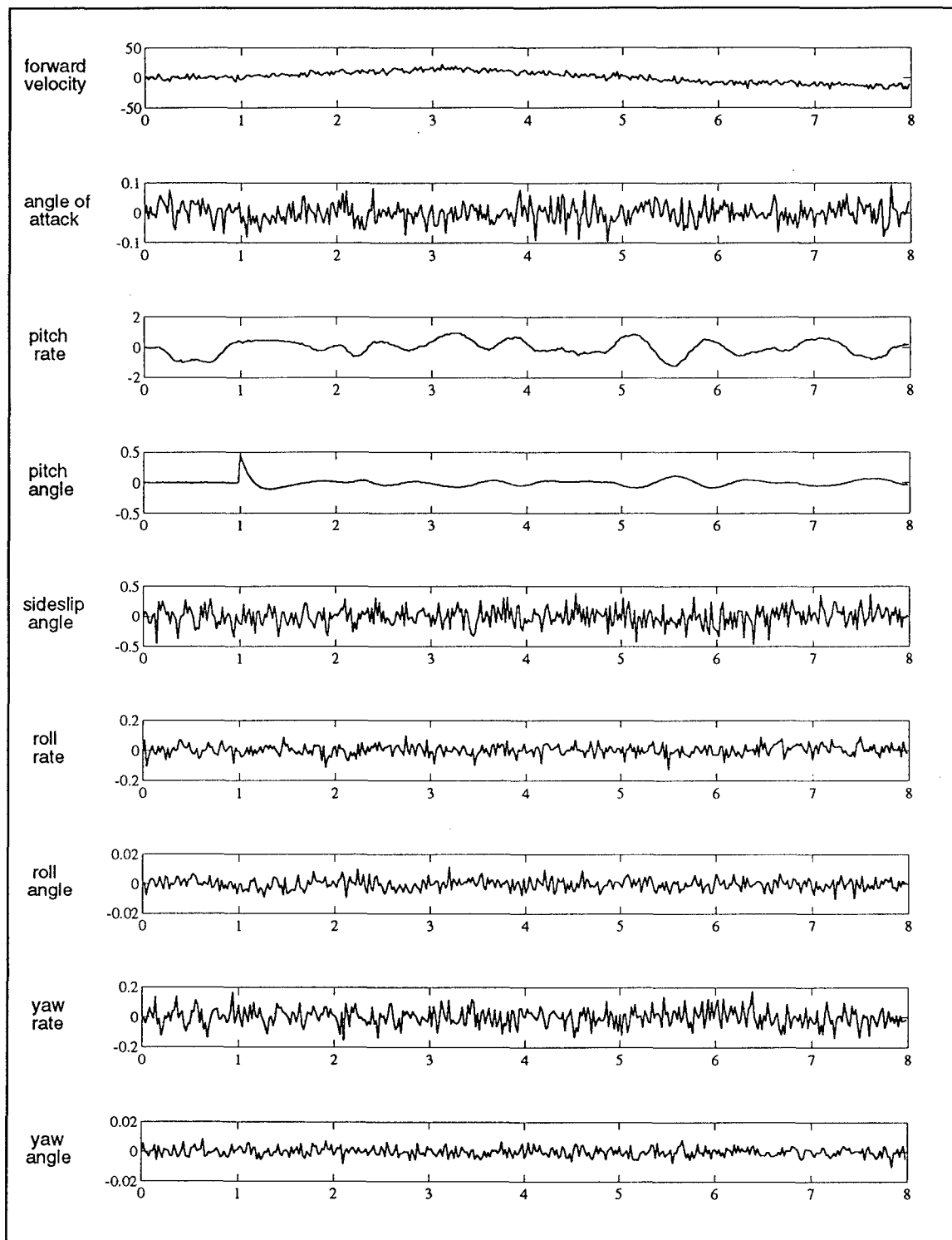
**Figure 22.** The Left Aileron Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case.

Table 10

Temporal Average of the Difference Between the Left Aileron Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Left Elevator Failure Test Case.

Residual Element	Temporal Average of Difference
Forward Velocity	$< 10^{-21}$
Angle of Attack	$< 10^{-21}$
Pitch Rate	$< 10^{-21}$
Pitch Angle	$< 10^{-21}$
Sideslip Angle	-1.255 E-18
Roll Rate	-6.940 E-18
Roll Angle	-5.321 E-17
Yaw Rate	-1.931 E-17
Yaw Angle	-3.987 E-17

4.3.3 Mismodeled Output Matrix (Sensor Failure). Finally, we examine the test case in which the true system has a sensor failure, specifically the pitch angle sensor, and compare the actual residual from the Kalman filter that assumed a pitch rate sensor failure with its equivalent residual computed using the residual from the Kalman filter that assumed a fully functional aircraft. Note that the actual failure, the assumed failure for the output filter residual, and the chosen source filter are nonmatching (three-way nonmatching) to help demonstrate the level of viability of this approach. We have chosen the source filter to be the fully functional filter to verify the structure shown in Figure 6 of Section 3.2.5. As in the previous cases, Figure 23 shows what appears to be a set of single plots of the fully functional Kalman filter residual, but these plots are actually the equivalent residual plotted over the actual residual. Once again, it appears to be a single



**Figure 23.** The Pitch Rate Failure Kalman Filter Residual and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual for a Pitch Angle Sensor Failure Test Case.

**Table 11**  
Temporal Average of the Difference Between the Pitch Rate Failure Kalman Filter Residual  
and the Equivalent Residual Computed Using the Fully Functional Kalman Filter Residual  
for a Pitch Angle Failure Test Case.

Residual Element	Temporal Average of Difference
Forward Velocity	-2.194 E-15
Angle of Attack	3.323 E-19
Pitch Rate	-3.686 E-19
Pitch Angle	4.966 E-18
Sideslip Angle	$< 10^{-21}$
Roll Rate	$< 10^{-21}$
Roll Angle	1.169 E-19
Yaw Rate	-2.575 E-21
Yaw Angle	-3.287 E-21

plot because the two residual are equivalent. The differences are tabulated in Table 11, and again demonstrate the equivalency.

**4.3.4 Overall Equivalency.** The average difference for all 4096 test cases was computed and tabulated in Table 12. These values indicate that the equivalent residuals are identical (within round-off error) to the actual Kalman filter residuals.

**4.3.5 Generalized Likelihood Ratio Class Equivalence.** We have [39] noticed the striking similarity between the filter "bank" structure of Figure 6 and Generalized Likelihood Ratio (GLR) based failure detection algorithms. The GLR structure [66] uses a Kalman filter residual as the input to a set of matched filters, and then the output of these matched filters are used by a hypothesis testing algorithm to determine the failure status. The matched filters are based

**Table 12**

Overall Temporal Average of the Difference Between an Actual Residual and the Equivalent Residual, Averaged over all 4096 Test Cases.

Residual Element	Temporal Average of Difference
Forward Velocity	9.633 E-17
Angle of Attack	-1.378 E-18
Pitch Rate	-1.518 E-17
Pitch Angle	7.768 E-18
Sideslip Angle	1.172 E-18
Roll Rate	2.686 E-19
Roll Angle	2.730 E-18
Yaw Rate	-3.139 E-18
Yaw Angle	1.576 E-18

on the designer's expectation of the time history of the residual for certain failures. For example, the set of filters would model a no-failure hypothesis, a jump in the residual mean (representing a sudden additive bias), and a ramp in the residual mean (representing a slow changing additive bias).

The results from this section have shown that the linear transforms of Figure 6 are equivalent to implementing a complete Kalman filter. Another interpretation of this result is that the linear transforms operating on a single Kalman filter output can be viewed as the best possible "matched filters" for the GLR failure identification method. These linear transforms yield the actual impact of the hypothesized failure on the Kalman filter residual, versus the designer's ad hoc representation of what that impact might be. This interpretation establishes an equivalence class relationship between MMAE failure detection and isolation systems and the best-possible GLR failure detection and isolation systems. This equivalence class relationship needs to be established

using a more structured theoretical approach. This appears as one of the recommendations for future research, in Section 5.6.

#### 4.4 Failure Detection Performance

We present the failure identification performance for the three MMAE structures that were developed in Chapter 3. The first structure, shown in Figure 24, is the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA). The Kalman filter equations for the SKFB were presented in Section 3.3.1, and the SHTA algorithm was presented in Section 3.3.1. An alternative structure, shown in Figure 25, was developed in Section 3.2.5 and was shown to be equivalent to the SKFB in Section 4.3. Since the structures are equivalent, the failure identification

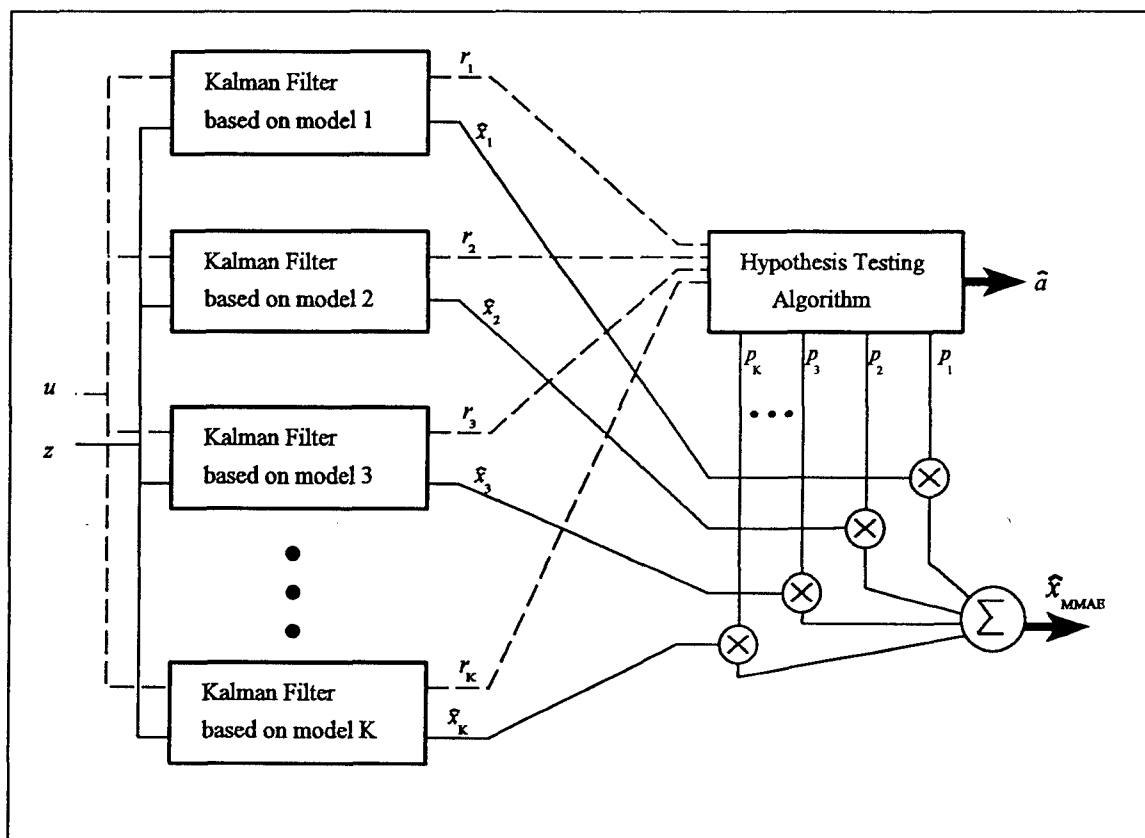
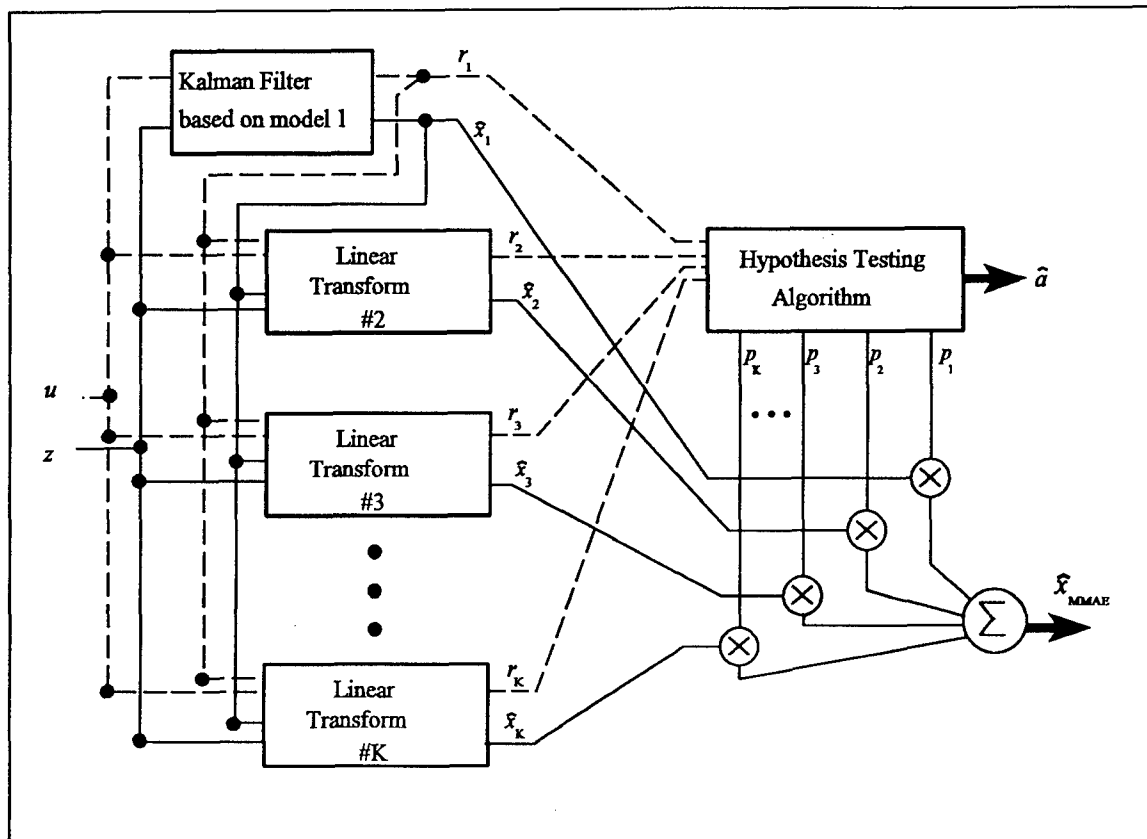
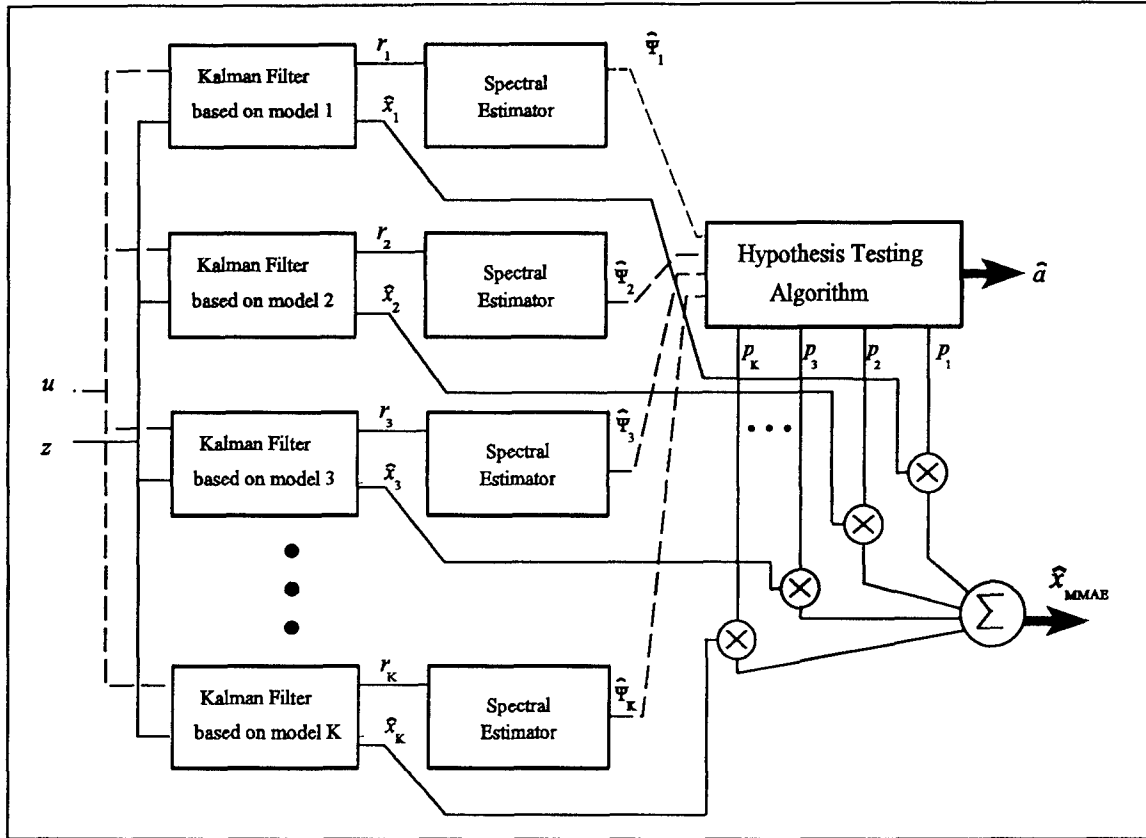


Figure 24. Multiple Model Adaptive Estimation Algorithm



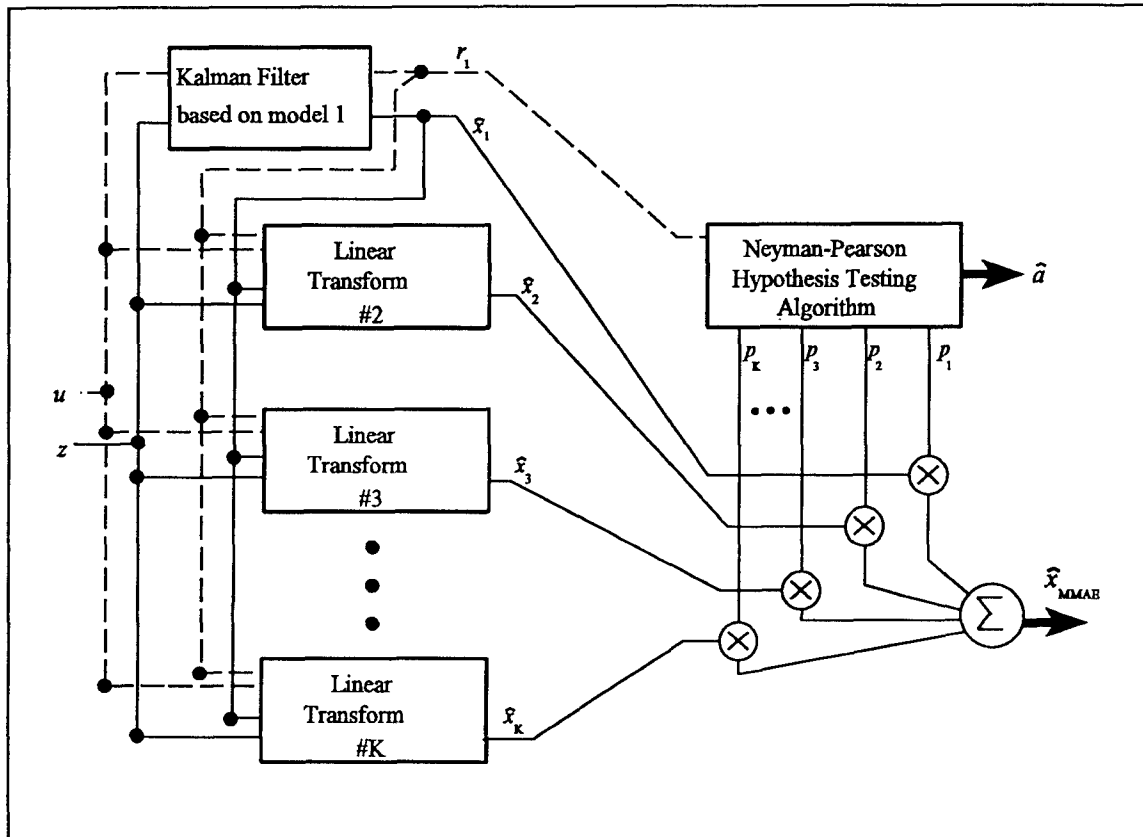
**Figure 25.** Alternative Multiple Model Adaptive Estimation Filter Bank Structure Using Equivalent Residuals

performance of the first structure will be identical to the failure identification performance of the equivalent structure. Therefore, we did not test the failure identification performance of this equivalent structure. The second structure, shown in Figure 26, is the Residual Correlation Kalman Filter Bank (RCKFB), developed in Section 3.2.6, with a SHTA. The last structure is the SKFB with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA) that was developed in Section 3.3.2. In Section 3.3.2.1, we showed that performing a hypothesis test, using a likelihood ratio, on multiple residuals was equivalent to performing multiple hypothesis tests on a single residual. Thus, we can replace the  $K$  filters in the Kalman filter bank, with a single filter and perform a Neyman-Pearson Hypothesis Test on the residual from this single filter. If the MMAE weighted state estimate is



**Figure 26.** Multiple Model Adaptive Estimation Algorithm using a Residual Correlation Kalman Filter Bank

needed, we would need to implement either the full SKFB, or the equivalent filter bank so that we have all of the state estimates from the various Kalman filters. Thus, the SKFB-NPHTA structure is equivalent to the structure shown in Figure 27. Since we do not need the state estimates to test the failure identification performance of this structure, we did not implement the linear transforms. We chose to construct the SKFB-NPHTA structure using the fully functional aircraft model based Kalman filter. The failure identification performance of the SKFB-SHTA is presented in Section 4.4.1, the RCKFB-SHTA performance in Section 4.4.2, and the performance for the SKFB-NPHTA is shown in Section 4.4.3. Section 4.4.4 is a comparison of the failure identification performances for these three structures. The last possible combination of these structures, namely the RCKFB-



**Figure 27.** Equivalent Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm Structure

-NPHTA, is a viable (and promising) algorithm, but it was not evaluated because of time limitations.

In Section 3.2.4 we found that the mean of the residual will have elements of the input for the case of an actuator failure, and for a sensor failure the residual will have elements of the state estimates. If we use a sinusoid for the input, then the sinusoid input elements cause the residual to be a sinusoid. This sinusoid appears as a spike in the power spectral density at the frequency of the input, which was shown in Figure 9. Therefore, since we can stipulate the frequency of the system input (since we can generate the dither input ourselves), we can use the spectral content at that frequency to indicate the presence of an actuator failure, which was the basis for developing the RCKFB. However, we cannot readily stipulate the frequency of the state estimates, unless extensive

simulation is done to find the particular input that causes sinusoidal state estimates. Since the state estimates are not necessarily sinusoidal, they may not be well correlated and we might need to estimate the power spectral density across the entire spectrum to try to find indications of the presence of a sensor failure. Figure 18 shows a residual from a filter with mismodeled output matrix. Clearly, the components of this residual are not sinusoidal. We believe that this technique would not work well for sensor failures unless the appropriate input is used that causes the states estimates to become strongly correlated in a readily distinguishable manner.

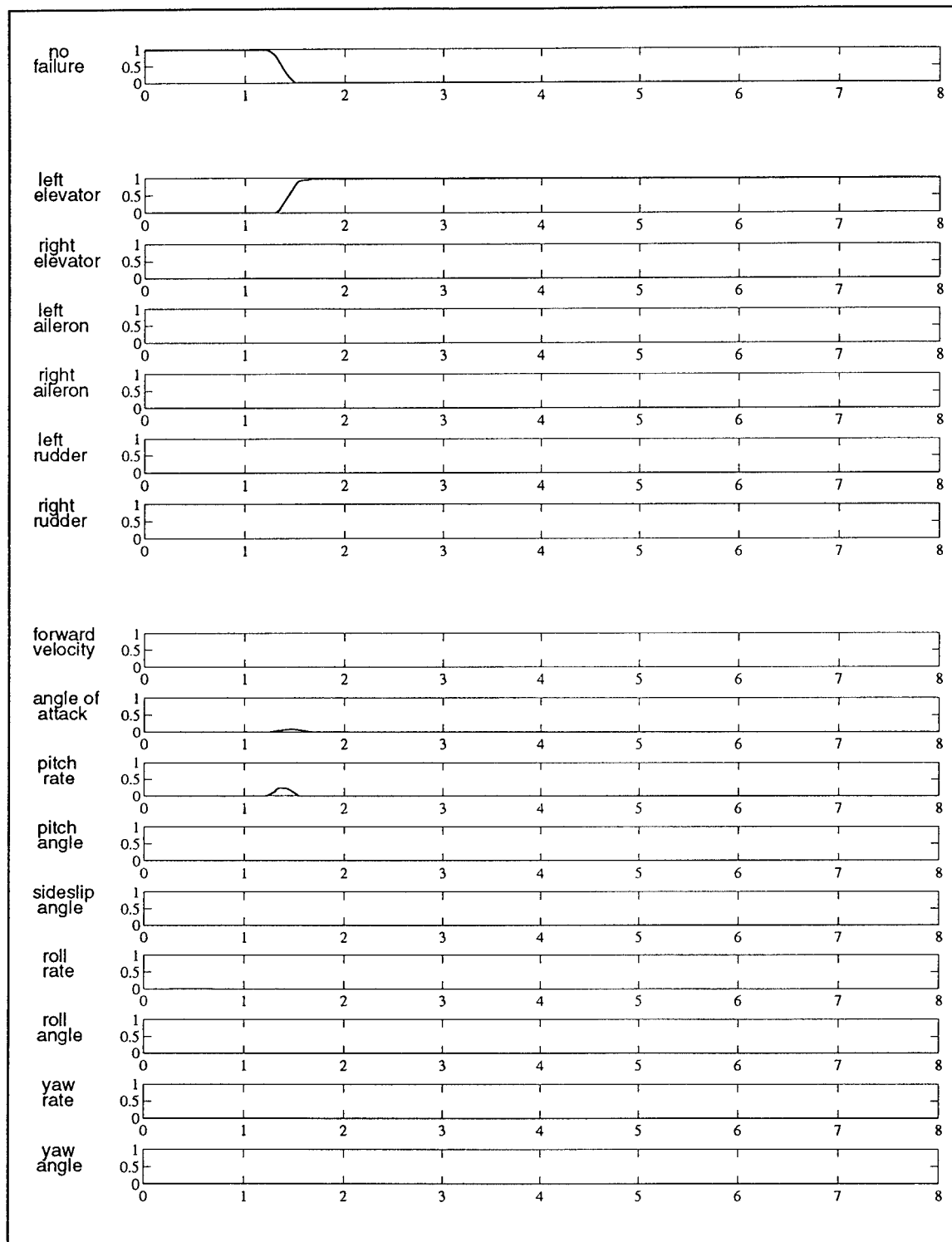
For this reason, we implemented the RCKFB-SHTA structure using only actuator failure models. The failure identification of the other structures was initially evaluated for all the hypothesized failures (including sensor failures), and then the set of hypotheses was reduced to just the actuator failures so we could compare the failure identification performance between the various structures. This reduced set of hypotheses significantly reduced the computations required to analyze the failure identification performance, so we were able to compare the failure identification performance of the various structures at several different system input levels. We also restricted the SKFB-SHTA and SKFB-NPHTA (to be presented in Section 4.4.1) structures to actuator failure hypotheses only, to make a fair comparison of failure identification algorithms. The usual implementation of these structures includes sensor failure hypotheses.

Thus, in the following sections you will first find the failure identification performance of each of the structures, and then a comparative evaluation of their performances for the different types of actuator failures. In their individual failure identification performance sections, we present the overall failure identification performance of the SKFB-SHTA and the SKFB-NPHTA, including sensor failures, but only the actuator failure identification for the RCKFB-SHTA. The actuator failure identification performance as a function of input level is presented for all three structures, and

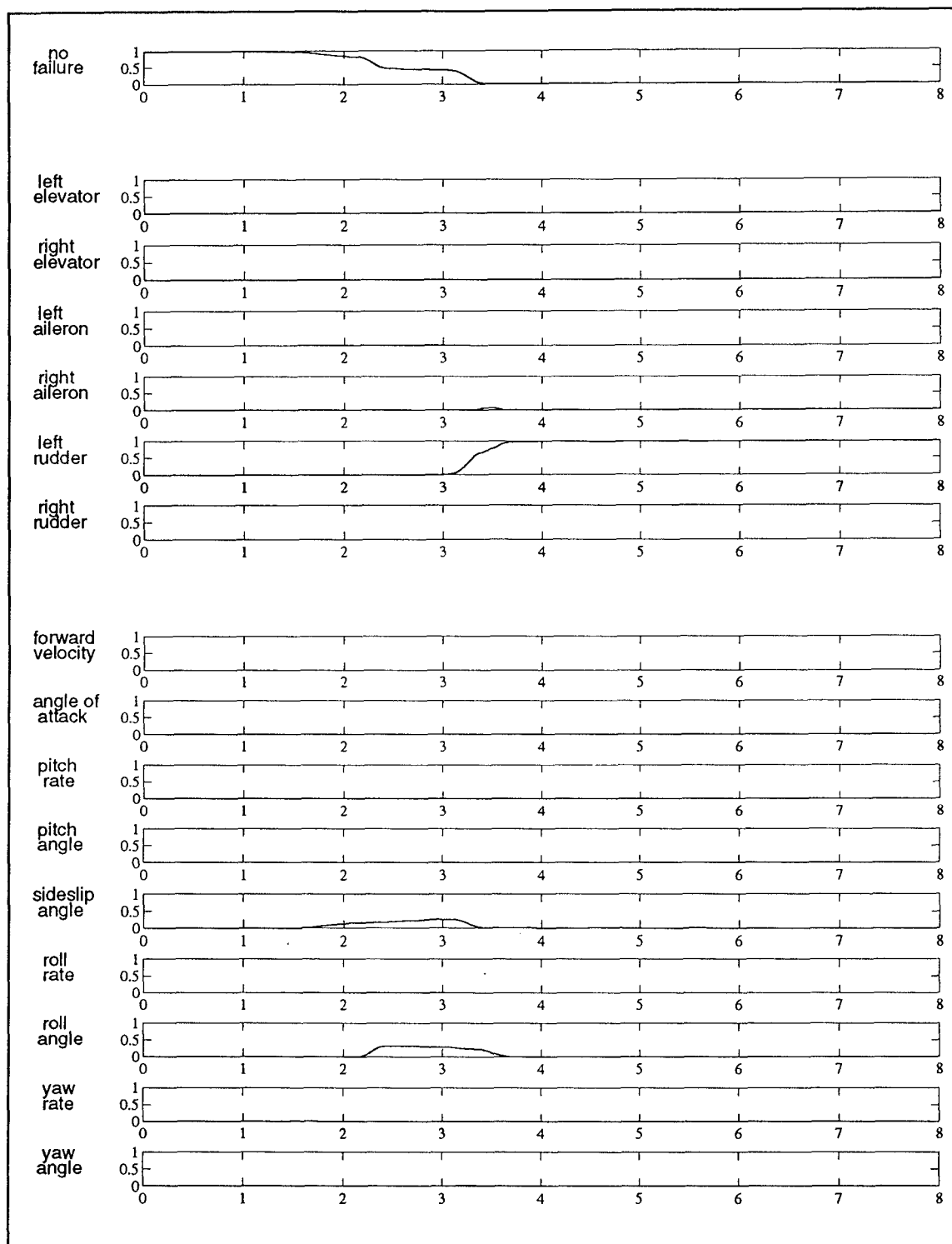
then the presentation of the actuator performance is rearranged to facilitate the comparison of the performance of the various structures.

4.4.1 SKFB-SHTA Failure Identification Performance. The failure identification performance of the SKFB-SHTA structure for this particular application was previously researched [19, 41] and some optimization of the modifications presented in Section 3.3.1 was accomplished. We present a summary of those results to facilitate performance comparisons to the other structures.

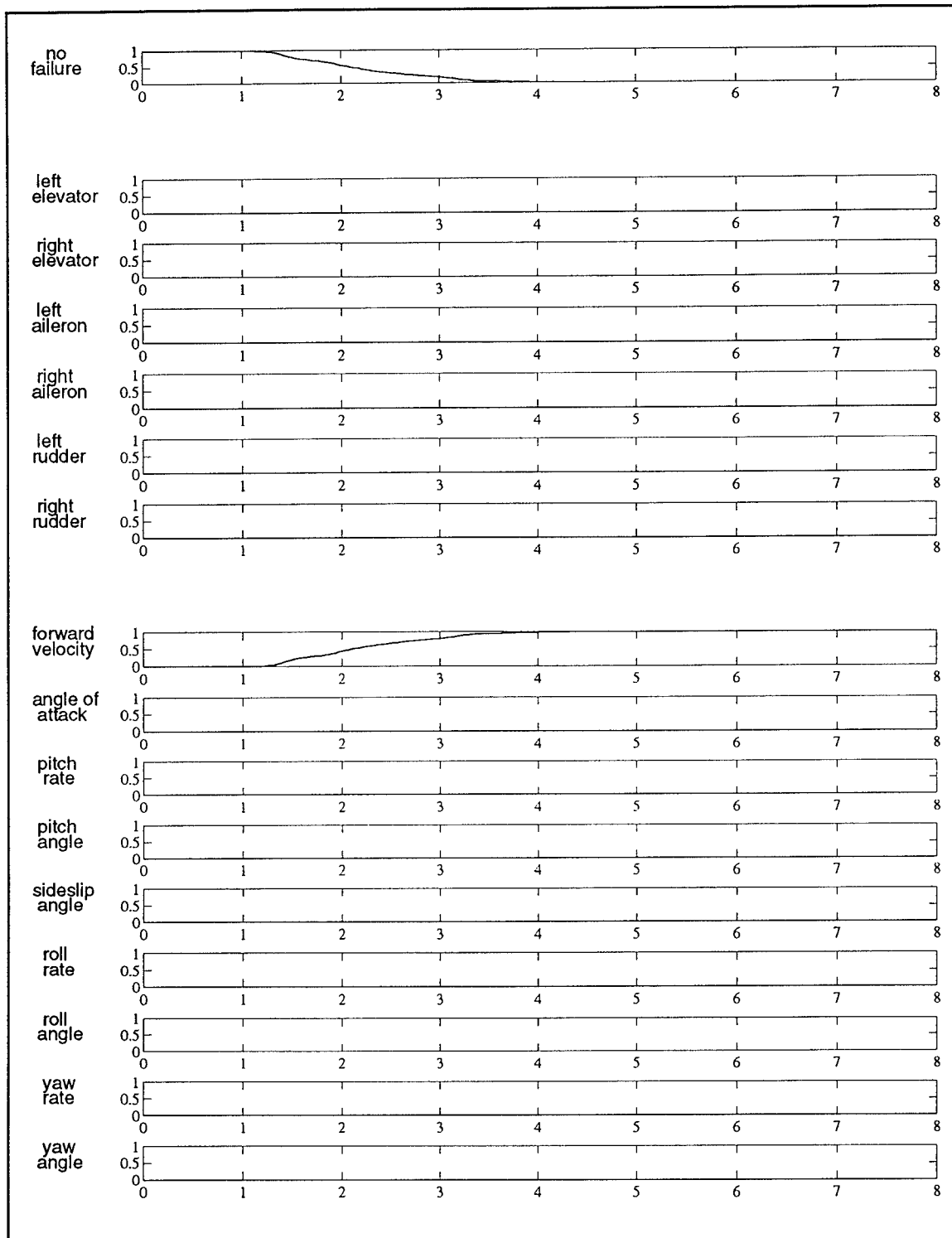
The failure identification performance for a particular failure (a left elevator failure) is presented in Figure 28. Note that the conditional probability for the fully functional hypothesis essentially mirrors the conditional probability for the actual failure (the left elevator failure). Similar results were observed for all other hypothesized failures, except for two. The failure identification was fairly slow for rudder failures (the left one is shown in Figure 29) and the forward velocity sensor failure (shown in Figure 30). We present these failures to show two reasons for slow failure identification performance. The rudder failure identification is slow because several other hypotheses appear to be equally as viable as the rudder failure hypothesis. Note that the roll angle sensor failure and sideslip angle sensor failures appear to be almost as probable as the fully functional hypothesis, until the rudder failure hypothesis is identified. Previous research [19] has shown that, unless the filters are properly tuned as we described in Section 3.3.1.4, this structure has a bias toward occasionally generating a false alarm for sensor failures, which is precisely what is observed in this case. The forward velocity sensor failure performance, Figure 30, shows a different situation. This sensor has much stronger measurement noise than all the other sensors, so the effects of this particular sensor failure have to be quite large for the failure to be identified. Figure 17 shows



**Figure 28.** Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Left Elevator Failure.



**Figure 29.** Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Left Rudder Failure.

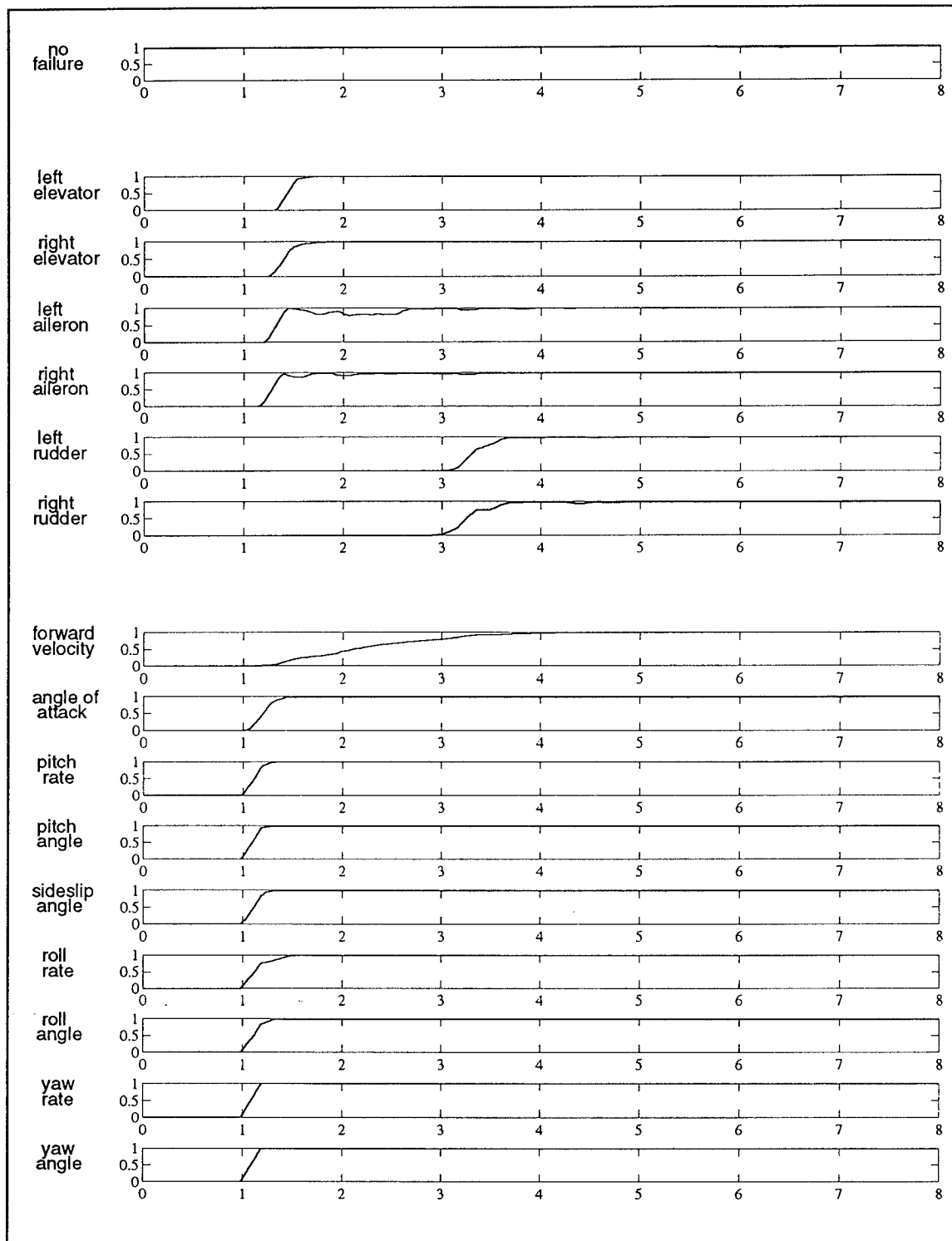


**Figure 30.** Failure Identification of the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA) for a Forward Velocity Sensor Failure.

that it takes a few seconds for the effects of the sensor failure to cause the residual to grow. In this case the noise of the sensor is so great that it masks the residual growth until it is quite large.

The results from these three and all other failure cases can be summarized by observing only the conditional probability of the particular failure that is actually occurring, since the fully functional hypothesis mirrors this probability, and all other hypothesized failures are near zero throughout the simulation. The overall performance of the SKFB-SHTA failure identification performance can be summarized by showing the conditional probabilities for each of the failure hypotheses on a series of plots, shown in Figure 31, where each plot shows the SKFB-SHTA failure identification performance for a particular failure. Note that the left elevator failure identification performance (Figure 28) is summarized in the left elevator plot of Figure 31, the left rudder failure identification performance (Figure 29) is shown in the left rudder plot of Figure 31, and the forward velocity sensor failure identification performance (Figure 30) is shown in the forward velocity plot of Figure 31.

One simple method of declaring a failure status is to choose a probability threshold that is used as a declaration trigger. The conditional probabilities are compared to this threshold, and when the threshold is exceeded, the hypothesis corresponding to that conditional probability is declared as the failure status. For example, if the threshold was set to a probability of 0.5 and a left elevator failure occurred, when the left elevator conditional probability exceeds 0.5, then a failed left elevator would be declared. Figure 31 shows that this would occur at about 1.5 seconds into the simulation, or about 0.5 seconds after the failure actually occurred. Using this threshold, we averaged the actuator failure performance of this MMAE structure over 5 Monte Carlo runs, and then averaged the left and right actuator failure performance results to get an average over 10 test samples. We averaged the left and right actuator failure performances because we assume that they are independent. This assumption is based on the orthogonal system input (Figure 13), where the left



**Figure 31.** Overall Failure Identification Performance of a Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA).

actuator input is almost exactly out-of-phase with the right input (a slight offset from exactly out-of-phase was needed to allow some excitation of the sensors), thus the left and right inputs are orthogonal. The averaging was done for a various input strengths, starting with the one shown in Figure 13, and reducing it with an input multiplier, down to 1/100 of the original control input. The results are tabulated in Table 13, where the decision time is the amount of time from the occurrence of the failure to when the conditional probability of the hypothesis crosses the probability threshold of 0.5. At very small inputs, the conditional probability grew so slowly, that when they approached the threshold, the randomness of the probabilities would cause it to cross the threshold a few times. For these cases, the decision time is the amount of time from the occurrence of the failure to when the conditional probability crosses the probability threshold for the last time and stays there. These results are graphically shown in Figure 32, where the input multiplier is plotted on a logarithmic (base 10) scale.

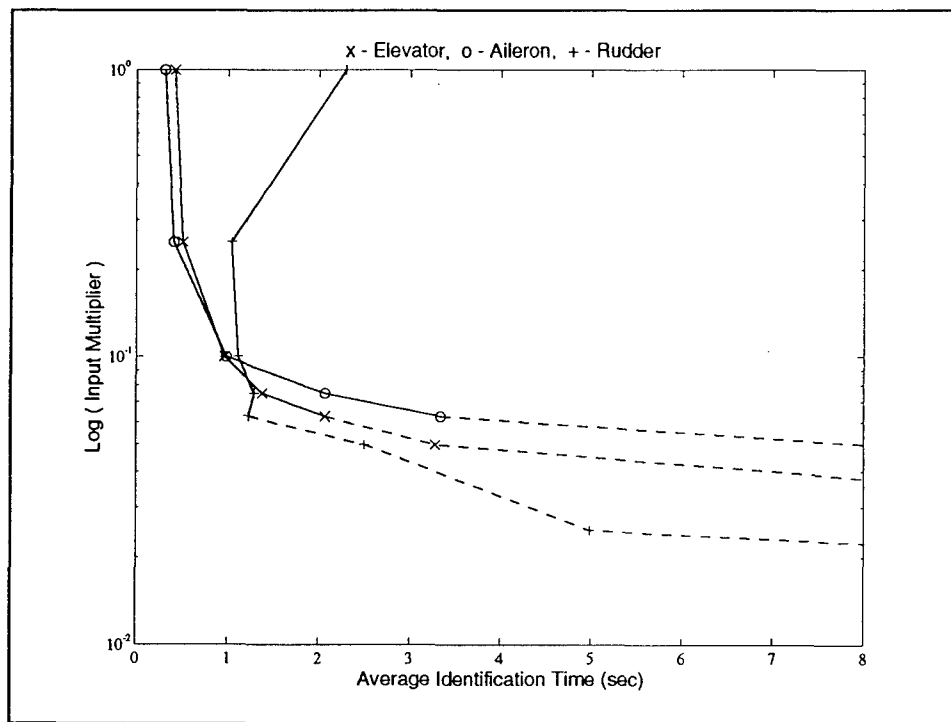
Note the "--" entries in Table 13 and the dashed line portion of the plots in Figure 32. These indicate where false alarms were observed, with an increase in the number of false alarms as the input was decreased. The dashes are to show where the performance may be deemed unacceptable due to the increase in the false alarm rate.

Figure 32 shows that failure performance drops off dramatically as the input decreases. This is particularly evident for the elevator and aileron failures, where identification times are beyond the termination of the simulation (7 seconds after the failure) for input multipliers of less than 0.025. These results make good sense, because stronger inputs would cause larger residuals from the incorrect Kalman filters when the failure occurs. Likewise, smaller inputs will produce relatively smaller residuals when the failure occurs, so the conditional probability of the failure hypothesis

**Table 13**

SKFB-SHTA Actuator Failure Identification Performance, 10-Test Sample Average.

Input Multiplier	Elevator Failures (sec)	Aileron Failures (sec)	Rudder Failures (sec)
1.00	0.42	0.31	2.29
0.25	0.51	0.41	1.04
0.1	0.96	0.98	1.11
0.075	1.38	2.07	1.29
0.0625	2.07 --	3.34 --	1.23 --
0.05	3.28 --	> 7 --	2.495 --
0.025	> 7 --	> 7 --	4.98 --
0.016	> 7 --	> 7 --	> 7 --
0.01	> 7 --	> 7 --	> 7 --



**Figure 32.** Actuator Failure Identification Performance for the Standard Kalman Filter Bank (SKFB) with a Standard Hypothesis Testing Algorithm (SHTA).

requires more time to grow. In particular, note that the rudder failure identification is still viable at an input multiplier level of 0.025, but drops off quickly after that.

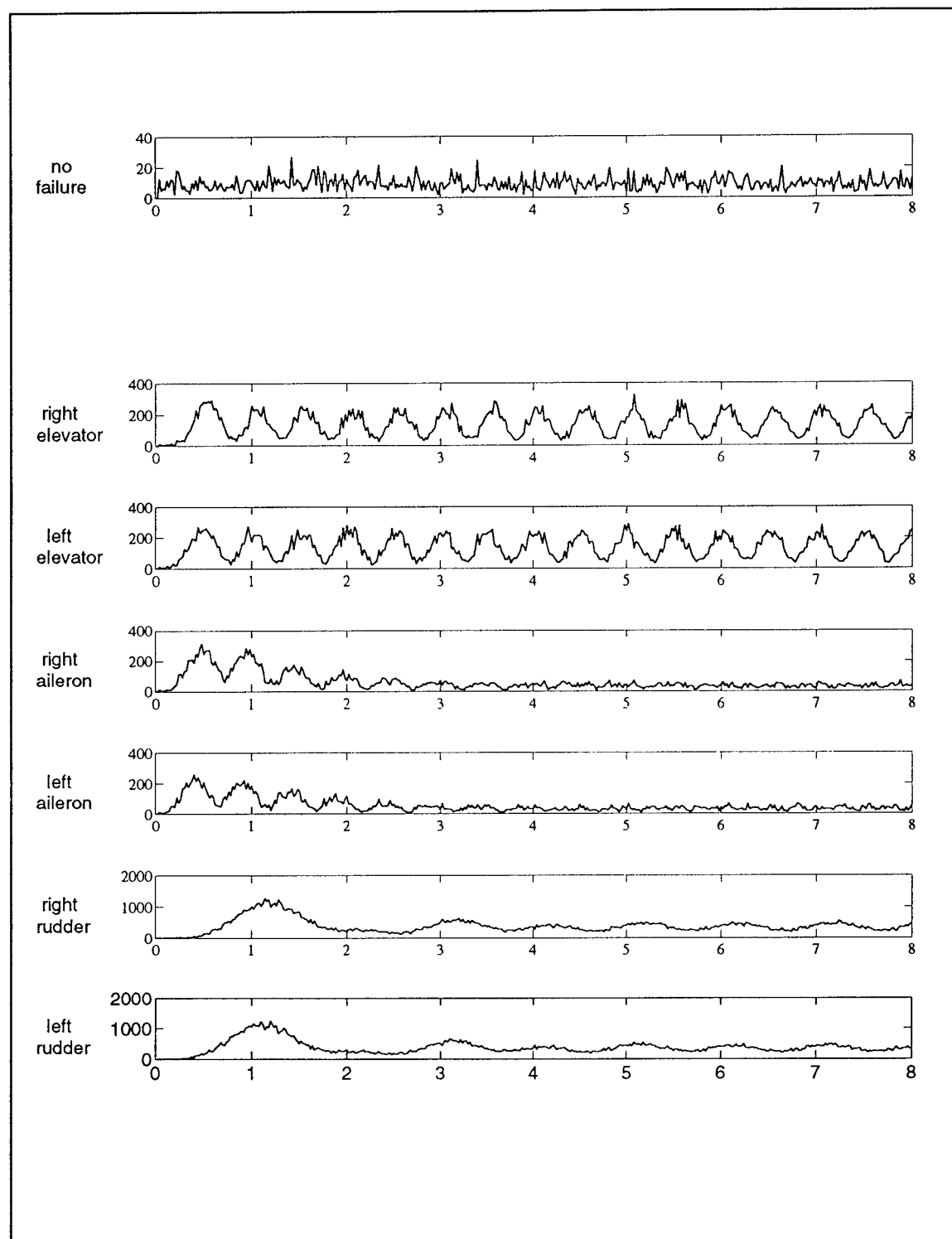
Note that, as the rudder input is increased, the failure identification time actually increases, which does not correspond to the anticipated results of better identification performance for stronger inputs. We showed earlier that this structure had trouble identifying rudder failures, possibly because of cross-coupling between axes causing misidentification of a sensor failure. This is not the case here, since, at this point, we have limited the SKFB to actuator failure hypotheses to make a fair comparison with the RCKFB structure. There are two reasons that this occurs, one is the stronger rudder input (Figure 13), relative to the elevator and aileron inputs, and the other is the timing of the failure.

To illustrate these reasons, we have plotted the likelihood quotient,  $q_k$  from Eq (99), from each of the Kalman filters for the no-failure case. These quotients are used as negative exponents in Eq (103), if  $\beta$  stripping described in Section 3.3.1.2 is not being used, and Eq (105), if  $\beta$  stripping is being used (we implemented the SHTA using  $\beta$  stripping for this research), and are key factors in computing the conditional probabilities. Large quotients produce extremely small exponential factors in Eq (103) and Eq (105), which, when multiplied by the prior conditional probability, tends to decrease the elemental conditional probability significantly. These elemental conditional probabilities are normalized by the sum of all the conditional probabilities. Good failure identification requires that the conditional probabilities change quickly when a failure occurs, which in turn requires that the quotient for the correct hypothesis be small compared to the other quotients, so that its conditional probability will not decrease as much as the others. This is clearly indicated in

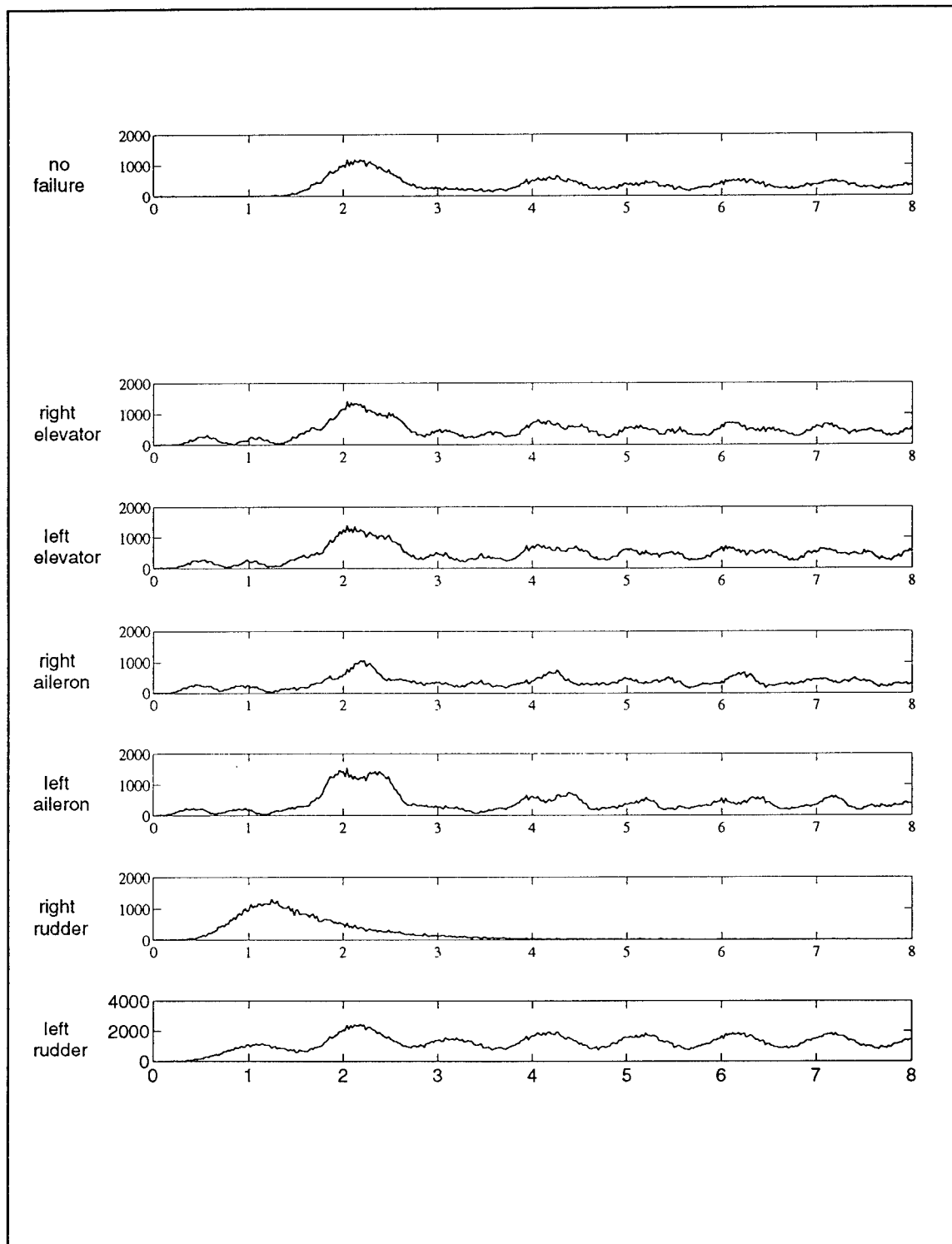
Figure 33, where the quotient for the no-failure Kalman filter is much smaller than the other quotients.

Using the plots on Figure 33, we see that, when the simulation is started, the strong rudder input causes the quotient in the rudder failure Kalman filter to grow quite large, much larger than the other quotients. When the right rudder failure actually occurs at one second into the simulation in Figure 34, the rudder quotient is nearly at its maximum, and almost five times larger than any other quotient. During the first second (i.e., with no failure), the large rudder input causes the state estimates to diverge rapidly from the true states, thus causing a large residual, which in turn causes a large rudder likelihood quotient. This divergence occurs in the other incorrect filters as well, but to a lesser degree due to the smaller inputs, which is why their likelihood quotients are smaller. When the failure occurs, the state estimates in the rudder failure Kalman filter (now the correct filter) take longer to converge to the true states than a filter whose state estimates were closer to the true states when it became the correct filter. Thus the residual will take longer to decrease, for the rudder failure case, which causes the likelihood quotient to decrease slowly, as shown in Figure 34. This causes the conditional probability to increase slowly, which delays the failure identification.

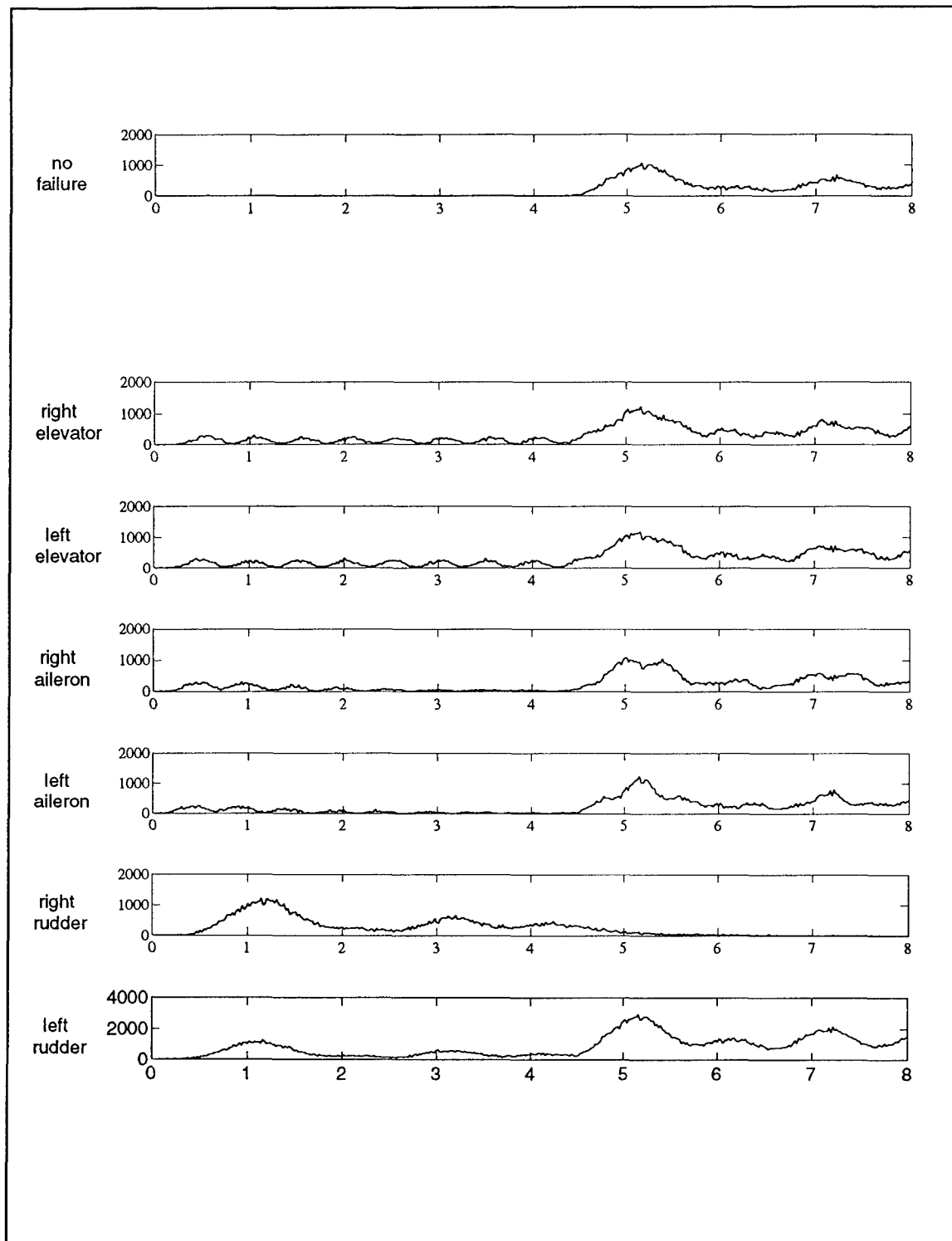
The timing of the failure is also important because of the interplay between the various likelihood quotients. Figure 33 shows that at one second into the simulation the rudder likelihood quotient is at its maximum and the other likelihood quotients are at a local minimum. Figure 34 shows that this causes a delay in the decrease of the rudder likelihood quotient, and the increase of the other likelihood quotients, which delays the change in the conditional probabilities. If the failure is injected at 4.2 seconds into the simulation instead, the rudder likelihood quotients (Figure 35) are approaching their local minimum, the aileron quotients are steady, and the elevator quotients are starting to rise to a local maximum. These conditions favor a faster rudder failure



**Figure 33.** Kalman Filter Likelihood Quotients for the No-Failure Case.



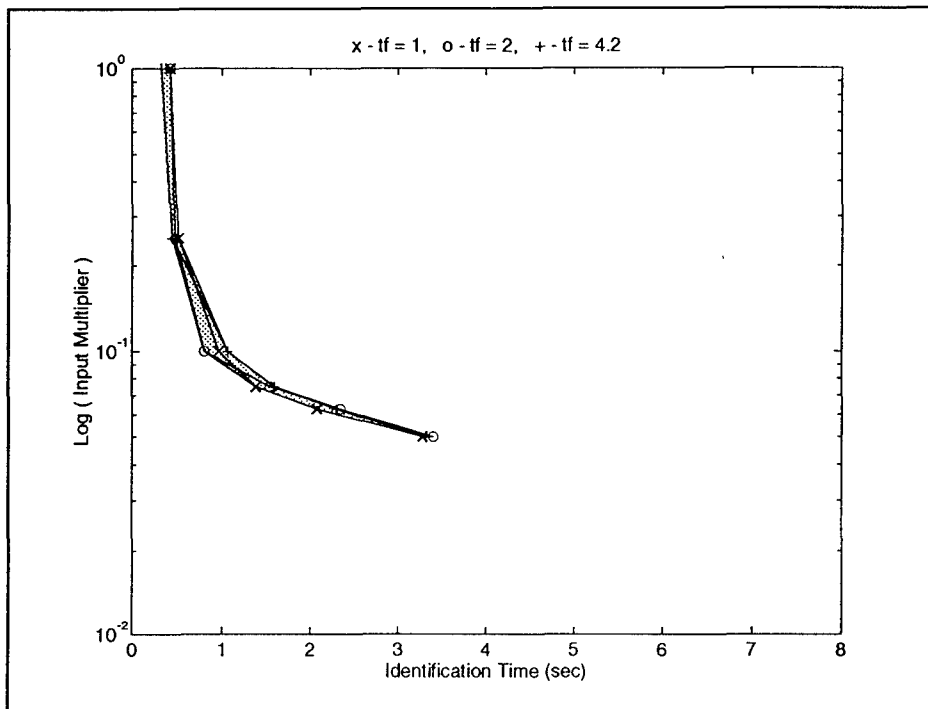
**Figure 34.** Kalman Filter Likelihood Quotients for the Right Rudder Failure Case.



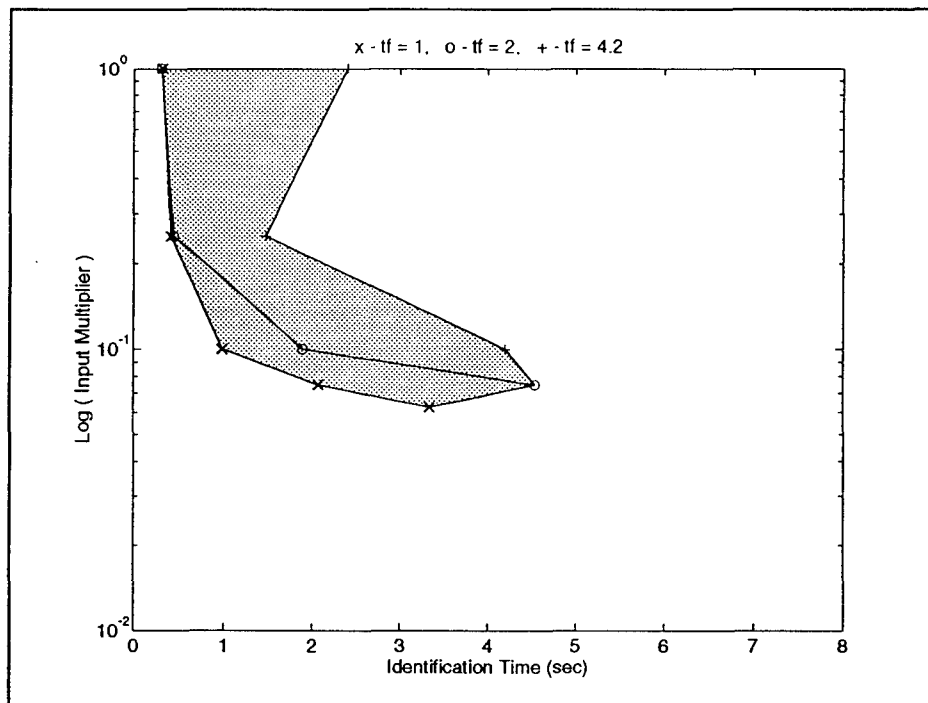
**Figure 35.** Kalman Filter Likelihood Quotients for a Right Rudder Failure Injected at 4.2 Seconds.

identification time because the rudder quotient is already diminishing, while the other quotients are increasing. On the other hand, these conditions appear to be unfavorable if an aileron failure were to occur at 4.2 seconds, since we found that the aileron failure detection time was much greater, under these conditions, than we found for the failure injection times of 1 and 2 seconds. We found that it took about 2.4 seconds to identify the failure clearly; during that time the aileron failure conditional probability and the fully functional conditional probability oscillated between each other, finally settling onto the correct hypothesis at about 2.4 seconds. Clearly, failure timing in relation to the likelihood quotients can strongly influence the failure identification performance.

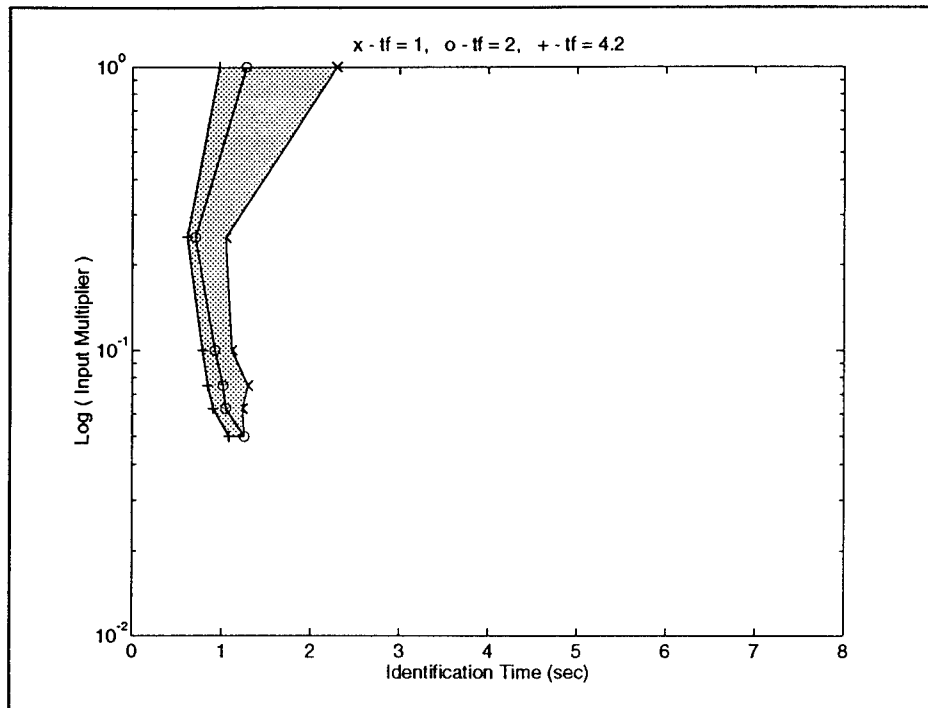
We conducted a the same type of analysis that was used to generate Figure 32, but we did this for different failure injection times. The failures were injected at 1, 2, and 4.2 seconds into the simulation. The input multiplier is plotted as a function of failure identification time for the elevator (Figure 36), aileron (Figure 37), and rudder (Figure 38). The shaded area represents the range of identification times that we believe could occur for the given input (shown in Figure 13), but with the different multipliers. Figure 36 indicates that there is an optimal input strength for which the difference between the maximum (worst) failure identification time and the minimum (best) failure identification time is minimized. This optimal input strength seems to be just prior to the dramatic dropoff in failure identification performance. This makes good sense because, as the input decreases, the relative strengths between the various inputs also decrease, so the relative strengths of the likelihood quotients are also decreasing. For example, let the inputs be 20, 10, and 15 degrees for the elevator, aileron, and rudder, respectively. If the multiplier is .25, then the inputs become 5, 2.5, and 3.75 degrees. If the multiplier is 0.1, then the inputs are 2, 1, and 1.5 degrees. Obviously, the range between the inputs is decreasing, which would cause a decrease in the range between the likelihood quotients.



**Figure 36.** Elevator Failure Identification Performance for Various Failure Injection Times.



**Figure 37.** Aileron Failure Identification Performance for Various Failure Injection Times.



**Figure 38.** Rudder Failure Identification Performance for Various Failure Injection Times.

Figure 36 shows excellent elevator failure identification performance with very little difference between the maximum and minimum identification times. We believe this is due not only to the strength of the elevator likelihood quotients, but also the oscillatory behavior that is clearly seen in Figure 33. This oscillatory behavior would help cause the likelihood quotient to build if the hypothesis is incorrect, and diminish quickly if the hypothesis is correct. These observations would argue for finding a "balanced" input, one that causes the oscillatory behavior seen in the elevator quotients in Figure 33, and cause equal magnitude likelihood quotients for all of the incorrect hypotheses, in order to get good detection performance for all types of actuator failures.

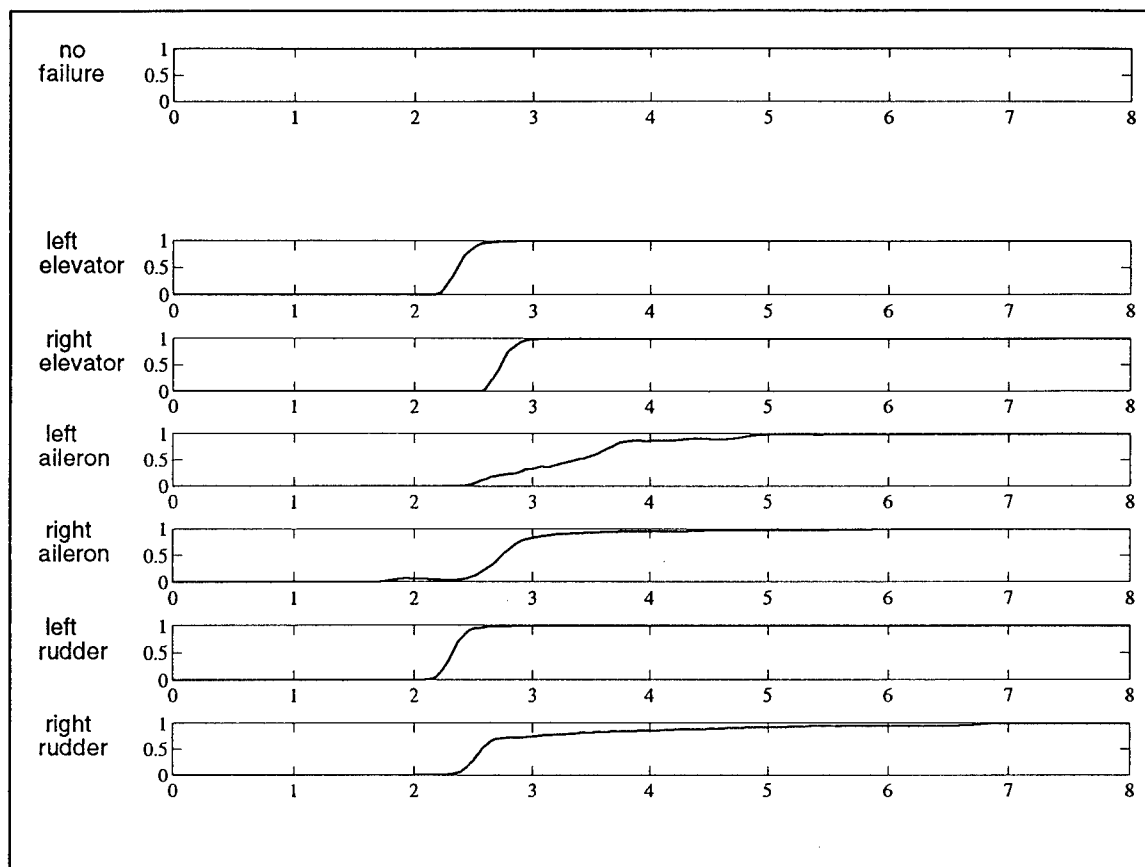
Time constraints prevented further analysis of these observations. In particular, we were not able to compare the performance of other structures for the different failure injection times, but we were able to compare the other structures' performance for the given input, shown in Figure 13.

The comparison of the failure identification performance for the various structures will be presented in Section 4.4.4.

**4.4.2 RCKFB-SHTA Failure Identification Performance.** We implemented the Residual Correlation Kalman Filter Bank (RCKFB) with the Standard Hypothesis Testing Algorithm (SHTA) shown in Figure 26. We used the form of the RCKFB, shown in Eq (72), that exploits the fast Fourier transform, but ignores the cross-coupling terms of the residual. These cross-coupling terms are not small in many cases. For example, the residual for a left elevator failure (Figure 14) shows that both the pitch rate and pitch angle strongly reflect the elevator input. Note that the pitch rate is clearly a sinusoid, and would be strongly correlated with itself. The pitch angle is also a sinusoid at the same frequency, so the cross-correlation between these two sinusoids would be quite strong. By ignoring these cases of strong cross-correlations, we are implementing an algorithm that does not exploit any of the information available from these cross terms. We considered the computational cost of exploiting these cross terms to be unacceptable, so we implemented the algorithm using only the autocorrelation of the individual elements of the residual.

A summary of the failure detection performance, following the format described in the previous section, is presented in Figure 39. Time limitations prevented investigating the sensor failure detection performance of this structure.

In Section 3.2.6 we noted that Kay [24] shows that this algorithm essentially filters the residual with a bandpass filter, centered at the frequency of interest and with a bandwidth of  $2/N$ . If the bandwidth of this filter is small, the other input frequencies will not have spectral content in the bandwidth of this bandpass filter. Thus, we need to collect enough data samples so that the bandwidth of the filter will not have any spectral content from the other inputs. The more data samples that we need to collect, the longer the delay will be from the time when the sinusoid is



**Figure 39.** Actuator Failure Identification Performance of a Residual Correlation Kalman Filter Bank (RDKFB) with a Standard Hypothesis Testing Algorithm (SHTA).

present (when the failure occurs) to when we can compute the spectral estimate of this filter and detect the sinusoid. Thus, there is a tradeoff between the number of data samples needed to distinguish between the input sinusoids and the failure detection time. This would dictate a wide spacing of the input sinusoids, to allow the bandpass filters to have larger bandwidths, so fewer data samples are needed. Unfortunately, most mechanical system (like aircraft) have very narrow low pass system response bandwidths (the bandwidth was about 2.2 Hz for this particular aircraft). The system response to the input must be seen in the residual for any failure identification scheme to work, thus we need to specify input frequencies that are within the system response bandwidth. This

requires that the inputs be located close together in the frequency spectrum, which requires narrow bandpass filters for this structure, which in turn requires more data samples to be collected. Our inputs were at 0.5, 1 and 2 Hz, which, when normalized by dividing by the sampling frequency, become 0.01, 0.02, and 0.04, respectively. The two closest frequencies were 0.5 Hz away from each other, or 0.01 in normalized frequency. The bandpass filter would need to have a bandwidth of less than 1 Hz (0.5 Hz on each side of the center frequency), which is 0.02 in normalized frequency, to be able to distinguish these input frequencies from each other. This requires 100 data samples ( $2/N = 0.02 \Rightarrow N = 100$ ) for the required bandpass filter bandwidth, which is the value of  $N$  chosen for this research. This value caused a delay in the detection of the failure because at least half of the 100 data samples (which translates to 1 second) had to show strong correlation at the input sinusoid frequency for the spectral estimator even to start to identify the presence of the sinusoid. Figure 39, when compared to Figure 31, shows that this delay caused this algorithm to take about one second longer to identify the actuator failures.

We noted earlier that the aircraft model assumed complete axis decoupling, which can help decrease the number of required data samples. The elevator input was at a frequency of 1 Hz, while the rudder input was at 0.5 Hz and the aileron input was at 2 Hz. Since the pitch axis would not affect the roll and yaw axes, the elevator input would not appear in the spectra for the longitudinal elements of the residual (roll and yaw). Thus the elevator input frequency did not need to be considered when we were designing the bandwidth of the bandpass filter. Thus, the two closest frequencies were actually 1.5 Hz apart, which means that we could have used only 34 data samples to distinguish between the rudder and aileron input frequencies. Time constraints prevented testing the failure identification performance using other values of  $N$ .

We tested the failure performance of this structure at the various input strengths that were used for the SKFB-SHTA testing. These results are tabulated in Table 14 and graphically presented in Figure 40. Note that the results are an average of only two Monte Carlo runs with both left and right actuators, for a total of four test samples. We observed a large drop in failure identification time for the strong rudder inputs. The reason for this phenomenon is illustrated in Figure 20. Note, when the failure occurs at one second, the pitch rate and pitch angle elements of the residual are at their peak and take about 0.5 seconds to die out. During this decay, these waveforms appear to be part of the sinusoid that was clearly present before the failure occurred. The sinusoid in the fully functional Kalman filter (Figure 21) is also delayed by about 0.5 second. Thus, the left elevator failure Kalman filter residual will show a slowly decaying spectral content at the elevator input frequency, and the fully functional Kalman filter residual will show a slowly building spectral content at that frequency. Thus, the SHTA will not identify the failure until the decay is nearly complete.

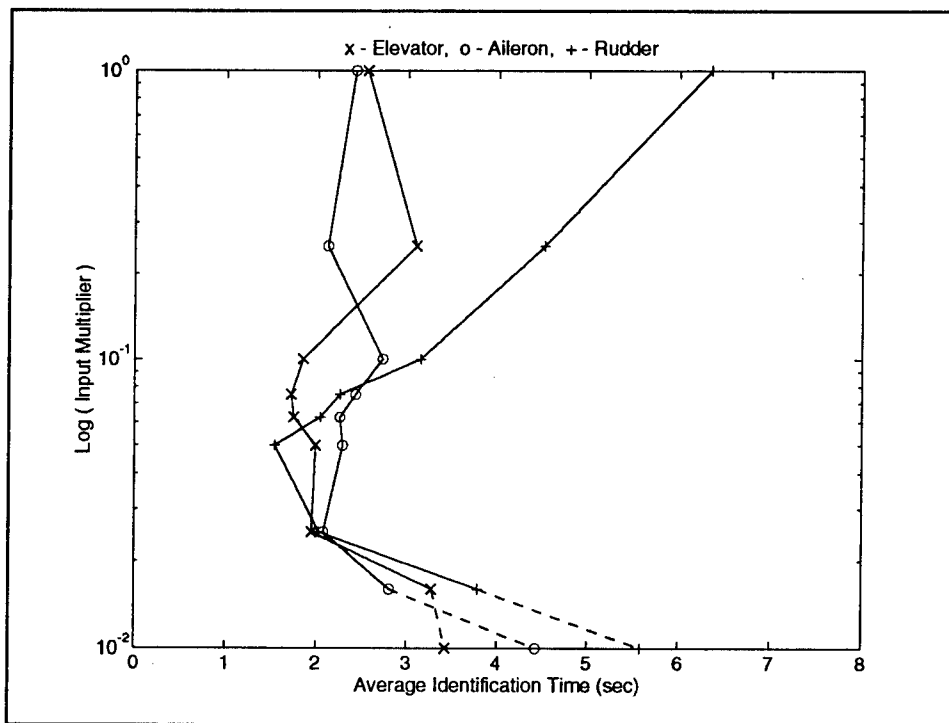
The important feature to notice in Figure 40, is that failure identification is still accomplished at a much lower input level than accomplished with the SKFB algorithm, as seen in Figure 32. This feature will be highlighted when this structure is compared to the others in Section 4.4.4.

The plots of the likelihood quotient, Figure 33, suggest an alternative structure to the RCKFB. We note that the likelihood quotient also reflects the sinusoidal nature of the residual, especially in the elevator likelihood quotients. A spectral estimator could be used to estimate the power spectral density of these likelihood quotients, rather than of the residuals. Since this quotient is a scalar that is formed by computing the scaled squared value of the residual, the cross-coupling terms would be included in this structure. Also, the spectral estimator would be fairly straightforward to implement, due to the decrease in the dimension of the input to the estimator.

Table 14

RCKFB-SHTA Actuator Failure Identification Performance, 4-Test Sample Average.

Input Multiplier	Elevator Failures (sec)	Aileron Failures (sec)	Rudder Failures (sec)
1.00	2.55	2.42	6.35
0.25	3.10	2.12	4.52
0.1	1.85	2.73	3.15
0.075	1.72	2.43	2.26
0.0625	1.75	2.26	2.04
0.05	1.99	2.29	1.54
0.025	1.95	2.08	2.03
0.016	3.28 --	2.81 --	3.79 --
0.01	3.43 --	4.43 --	5.59 --

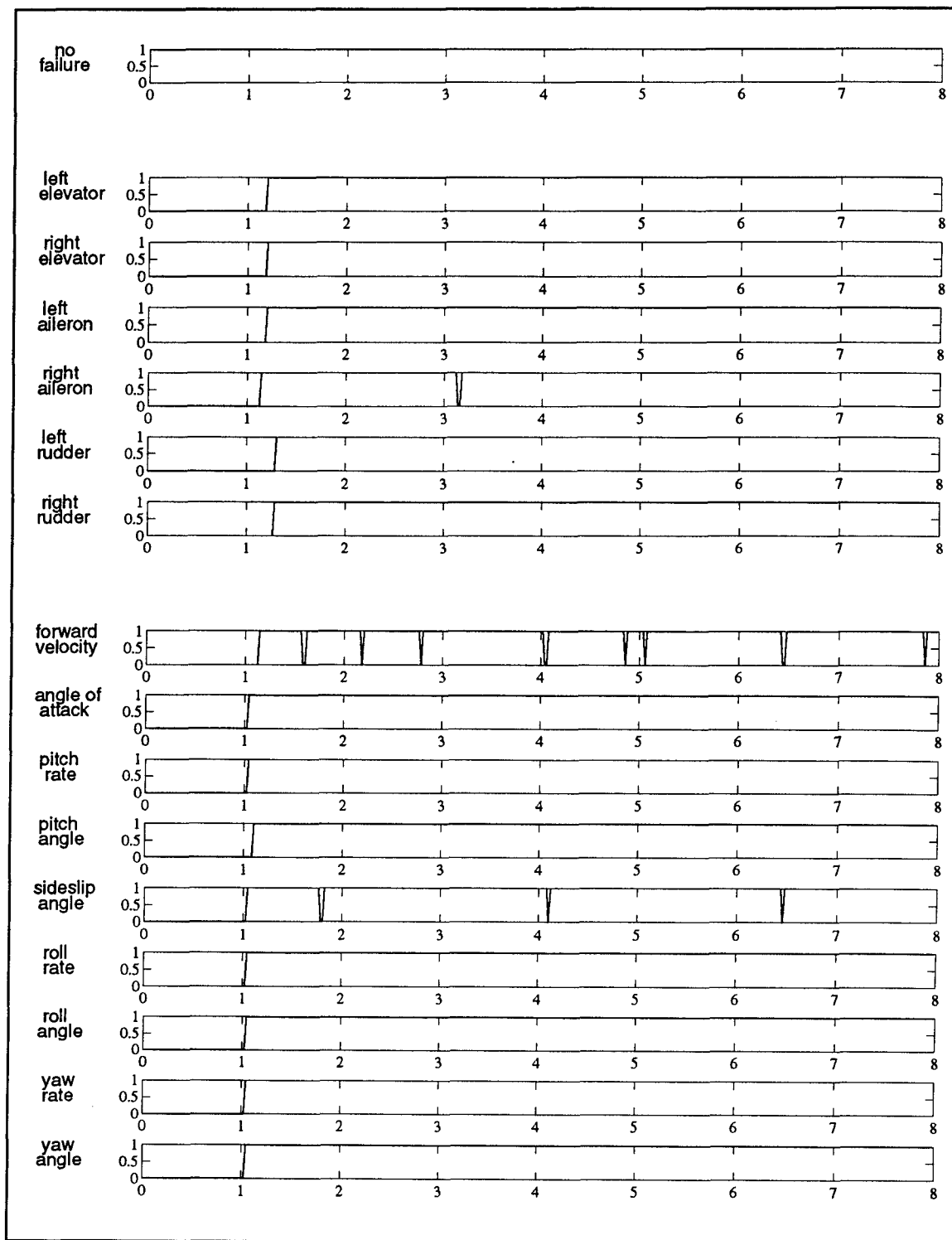


**Figure 40.** Actuator Failure Identification Performance for the Residual Correlation Kalman Filter Bank (RCKFB) with a Standard Hypothesis Testing Algorithm (SHTA).

Note, that by using this likelihood quotient (the scaled squared norm of the residual) we are essentially squaring the sinusoidal components of the residual, which doubles the frequency, so the likelihood quotient oscillates at twice the frequency as the residual. This would allow us to use fewer data samples since the bandwidth of the bandpass filters could be twice as large as they were for the RCKFB. This approach shows great promise for improving the performance of residual correlation based Kalman filter banks.

**4.4.3 SKFB-NPHTA Failure Identification Performance.** Using the development in Section 3.3.2, we implemented a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA), similar to the structure shown in Figure 27. We were only testing the failure identification performance of the structure, so we did not implement the linear transforms that are needed to produce the blended state estimate. Thus our structure simply consisted of a single Kalman filter (the fully functional Kalman filter) with a NPHTA. The output of this algorithm is a declaration of the failure status, not a set of conditional probabilities. We chose to compare the declared failure status to the true failure status to evaluate the performance of this algorithm. In Figure 41 we present the failure identification performance of this structure, by plotting the agreement (denoted by 1) and disagreement (denoted by 0) of the declared failure status with the true failure status.

The design parameters for the NPHTA are the probability of detection and probability of false alarm, which were 0.999 and 0.01 respectively, for this research. Performance sensitivity to the choice of these two parameters could be investigated for a given application, but these are chosen as reasonable, representative values. First note the one misidentification in the right aileron plot of Figure 41. The fully functional hypothesis was momentarily (for about 0.02 sec) declared as the failure status, when in fact the right aileron failure was the true hypothesis. We attribute this to the probability of false alarm setting. Note that this is a false alarm (using the definitions of Section 3.3)



**Figure 41.** Overall Failure Identification Performance of a Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA).

rather than a missed alarm, since the primary hypothesis before the test was the failed state but the identification algorithm chose the unfailed state. From the time of failure (1 sec) to the end of the simulation there are 48 tests of the right aileron failure hypothesis vs. the fully functional hypothesis. The probability of false alarm setting that was used would roughly give us one false alarm for every 100 tests, so to get one false alarm for a set of 48 tests does seem reasonable.

Sensor failures showed mixed results. The failure of sensors that have very little measurement noise was identified quite readily, but the two sensors that have the most measurement noise show several false alarms. This algorithm was developed under the assumption that the covariance of the test statistic was the same for all of the hypotheses. This is true for the actuator failures, but not necessarily true for the sensor failures. If the sensors have very little measurement noise, then the covariance of the test statistic, for that particular hypothesis, is essentially equal to the covariance for an actuator failure hypothesis. If the sensor has a large amount of measurement noise, then the covariance matrix will differ greatly from the covariance matrix for an actuator failure hypothesis. We believe that the different covariances are the reason for the large number of false alarms for these sensor failure hypotheses.

The failure identification performance for this structure was tested using the range of inputs from the previous sections, and the results are tabulated in the Table 15. The plot of those results (Figure 42) shows that this structure performs quite well, especially when compared with the SKFB-SHTA, Figure 32, and RCKFB-SHTA, Figure 40 (except for small input strengths). Note that this algorithm identifies the failure much more quickly than the other structures, until the input reaches 0.05 times the original input, and then the performance drops off quite rapidly. We found an increase in false alarms for input levels below the 0.05 multiplier. It appears that the 0.05 multiplier level is the minimum input strength needed to provide good failure identification performance. Note

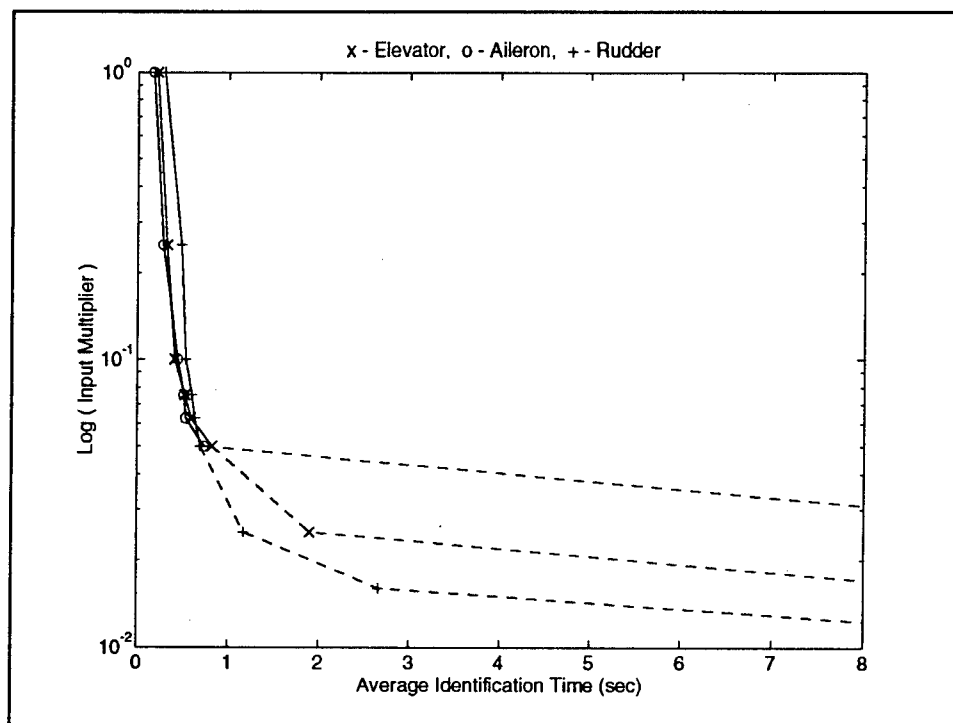
that this structure's performance remains relatively constant with input strength, until it reaches this minimum input.

In particular, note that the rudder failure identification performance does not reflect the drop in performance as the input strengthens, like we saw for the other two structures. The basis of the SHTA is that the correct residual is small compared to the residual from the incorrect filters. We've seen that, if the failure occurs when the appropriate residual is near a peak, the failure identification is delayed. This algorithm computes a time history of the residual, based on a certain hypothesis, and then compares the measurement history of the residual to the computed time history (if the distinguishability is great enough) to determine which hypothesis is correct. Thus, the algorithm does not have to wait until the residual decays before it can identify the failure. It is possible that this algorithm could suffer from the failure injection timing problems that were described for the SHTA. If the failure occurs just after the discrimination measure was reset, when the discrimination measure grows large enough to trigger a test, the test statistic will not have grown as large as it would have if the failure had actually occurred right when the discrimination measure was reset. Thus, the NPHTA could miss the change in hypothesis (missed detection) when the test is conducted. The discrimination measure and test statistic would be reset, and they would build appropriately to identify the failure on the next test. We observed that actuator failure hypothesis tests were triggered about every 5 - 10 time samples and sensor failure hypothesis tests were occurring about every 1-5 data samples. Thus, it is possible that the failure identification could be delayed about 0.2 seconds beyond the best detection time. This compares quite favorably with the range in detection times that were found using the SKFB-SHTA structure, shown in Figures 36, 37, and 38. A similar analysis of the timing differences needs to be accomplished for this structure, in the future.

**Table 15**

SKFB-NPHTA Actuator Failure Identification Performance, 10-Test Sample Average.

Input Multiplier	Elevator Failures (sec)	Aileron Failures (sec)	Rudder Failures (sec)
1.00	0.21	0.17	0.29
0.25	0.32	0.28	0.48
0.1	0.41	0.44	0.53
0.075	0.52	0.51	0.60
0.0625	0.59	0.53	0.63
0.05	0.82 --	0.73 --	0.69--
0.025	1.93 --	> 8 --	1.17--
0.016	> 8 --	> 8 --	2.66 --
0.01	> 8 --	> 8 --	> 8 --



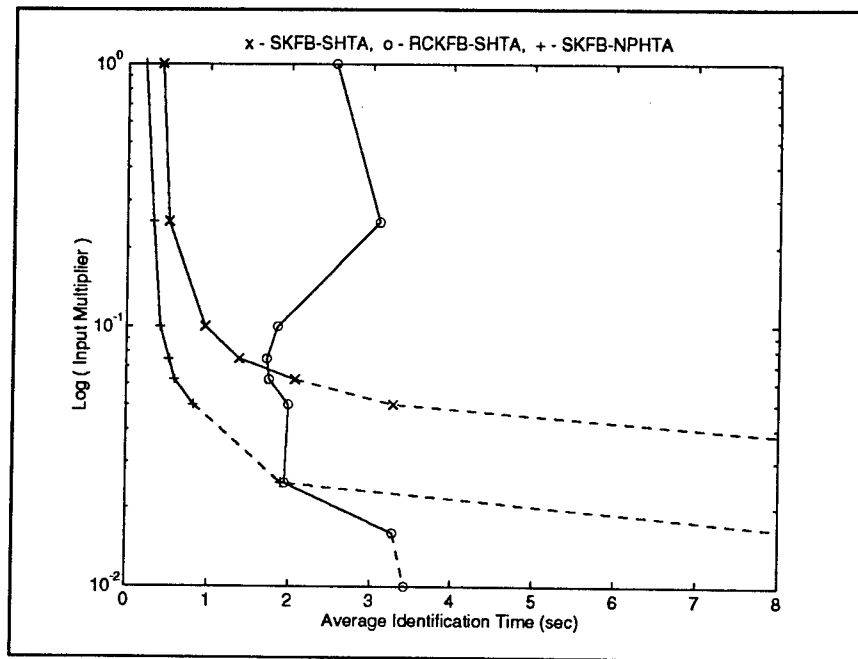
**Figure 42.** Actuator Failure Identification Performance for the Standard Kalman Filter Bank (SKFB) with a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA).

#### 4.4.4 Comparative Failure Identification Performance. We used the failure

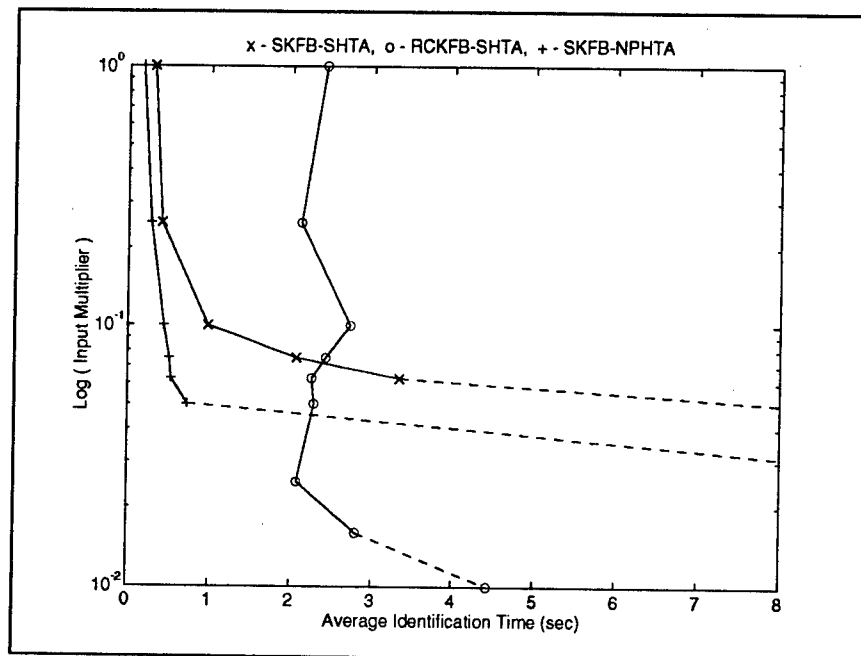
identification performance data shown in the last three sections to compare the performance for the various structures. The elevator failure identification performance of all three structures is shown in Figure 43. Similarly, the performance for an aileron failure is shown in Figure 44, and the rudder failure identification performance is shown in Figure 45.

Note that, for a strong input, the SKFB-NPHTA structure performs the best of the three structures. This was the only structure that did not show a decrease in failure identification performance for stronger inputs for the rudder failure. This structure identified the failures in about half the time that it took for the SKFB-SHTA structure, and about one tenth the time it took for the RCKFB-SHTA. The RCKFB-SHTA structure took longer at these input levels because 100 data samples needed to be collected and the spectral estimate computed, which delayed the failure identification. Collecting a small number of data samples at these high input levels might produce faster failure identification times.

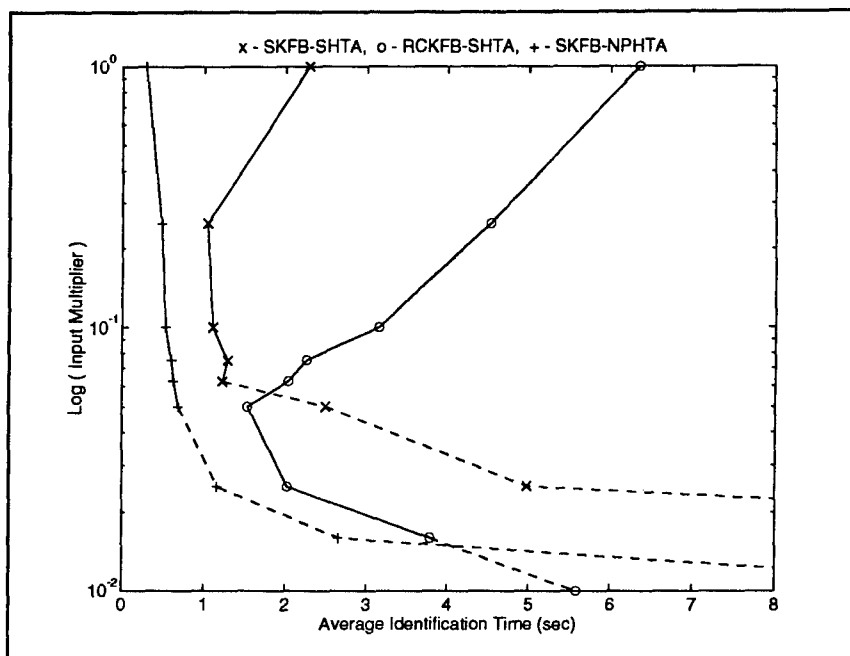
As the input decreases, the failure identification performance generally decreases for the SKFB-SHTA and the SKFB-NPHTA structures, and seems to remain fairly constant for the RCKFB-SHTA structure. The exception to this general rule is the decrease in rudder failure identification performance as the input strength increases for the SKFB-SHTA and RCKFB-SHTA structures. This was caused by both the relative strength of the rudder input, which caused a larger residual amplitude, and the timing of the failure, which occurred just as the residual peaked, so the residual took longer to decay. We were unable to conduct a failure injection timing analysis for the RCKFB-SHTA and the SKFB-NPHTA structures, but we believe that these structures would suffer the same problems. The SKFB-NPHTA does not show this behavior on these plots because the failure occurs just as the discrimination measure is reset. We believe that the SKFB-NPHTA would



**Figure 43.** Comparative Failure Identification Performance for an Elevator Failure.



**Figure 44.** Comparative Failure Identification Performance for an Aileron Failure.



**Figure 45.** Comparative Failure Identification Performance for a Rudder Failure.

have very small delays due to the timing of the failure injection. These conjectures need to be researched in the future.

At small inputs levels the performance drops of rapidly, especially for the SKFB-NPHTA. The RCKFB-SHTA provides the best failure identification performance of the three structures at low input levels. For input levels below 1/20 of the original input used for this research, the RCKFB-SHTA structure was the only structure that still provided failure identification performance.

All three structures have a minimum input level below which their performance diminishes rapidly. As the input strength is decreased, the approach of this minimum input level for good failure identification performance is signaled by an increase in the number of false alarms. This suggests that the false alarm rate could be monitored (by using statistical testing for estimating the false alarm rate) and this could be used as feedback to set the input level appropriately. We also found that the

differences in failure time identification for the SKFB-SHTA could be minimized by setting the input level just above the point where the failure identification performance starts to deteriorate.

These results strongly suggest that a combination of structures would provide the best failure identification performance. This combination would use the SKFB-NPHTA structure to provide good detection at strong input levels, and the RCKFB-SHTA structure at low input levels. Another possibility is the RCKFB with the NPHTA, which may provide the good failure identification performance at both strong and low input levels. Another possible structure would combine all three to provide redundancy, which could then be used to decrease the false alarm rate by using voting methods. Also, each of these structures could be implemented using equivalent residuals. This set of MMAE structures provides a rich assortment of possible combinations.

## 4.5 Chapter Summary

In this chapter we have presented the results of the performance evaluation of the algorithm structures that were developed in Chapter 3. We started by confirming that the development in Section 3.2.4 and 3.3.3, where we characterized the mean and covariance of the residual in the presence of mismodeling in the input control, output, and state transition matrices. We corroborated this characterization for mismodeling in input control and output matrices by making graphical and statistical comparisons. We then validated the development of the equivalent residuals in Section 3.2.5. We compared the failure detection performance for three different MMAE structures, the Standard Kalman Filter Bank (SKFB) with the Standard Hypothesis Testing Algorithm (SHTA), the Residual Correlation Kalman Filter Bank (RCKFB) with the SHTA, and the SKFB (which was just a "bank" consisting of a single Kalman filter) with the Neyman-Pearson Hypothesis Testing Algorithm (NPHTA). We found that the SKFB-NPHTA provided the quickest failure identification at fairly

strong system input levels, and the RCKFB-SHTA provided failure identification at system input levels far below the levels required by the other two structures.

## V. Conclusions

### 5.1 Chapter Overview

In this chapter we present our conclusions from the research results shown in Chapter 4. In Section 5.2 we look at the characterization of the mean and covariance of the Kalman filter residual, in the presence of mismodeling in the control input matrix, output matrix, and state transition matrix. We present our conclusions, in Section 5.3, concerning an equivalent structure to the Kalman filter bank in which the residual from a single Kalman filter, along with its state estimates, are used to compute the equivalent residual from a different Kalman filter. In Section 5.4, we draw some conclusions concerning a new version of the Kalman filter bank in which spectral estimates of the residuals are used by the hypothesis testing algorithm to make a failure status evaluation. Section 5.5 presents our conclusions for the Neyman-Pearson-based hypothesis testing algorithm. Finally, we close with our recommendations for future research.

### 5.2 Characterization of the Residual

We developed expressions for the mean and covariance of the Kalman filter residual in the presence of mismodeling in the input control matrix, output matrix, and state transition matrix, in Sections 3.2.3 and 3.2.4. These expressions were tested by comparing the computed mean and standard deviation to actual residuals in Section 4.2. The comparison was made for three specific cases: no mismodeling, a mismodeled control input matrix, and a mismodeled output matrix. We found that the computed mean and standard deviation accurately characterized the actual residual mean and standard deviation, thus verifying our development.

### 5.3 Single Residual Kalman Filter Bank

We developed a methodology in Section 3.2.5, in which the residual and state estimates from one Kalman filter can be used to compute the equivalent residual (and state estimates if needed) from a second Kalman filter, if the system model used by the second filter differs from the first filter's model, only in the input control matrix, output matrix, and state transition matrix. We propose an MMAE-like algorithm composed of an equivalent Kalman filter bank in which only one filter is fully implemented, and then that filter's residual and state estimate are used to compute the equivalent residuals from the other Kalman filters (that are not actually implemented) in the filter bank. In Section 4.3, we compare the equivalent residuals with the actual residuals, and found that they are indeed, equivalent. This structure has the potential to attain great computational savings, since only one Kalman filter is actually implemented.

We noted [39] an equivalence class relationship of this structure with the Generalized Likelihood Ratio (GLR) failure detection structure. The GLR failure detection structure uses a set of matched filters that operate on the residual from a single Kalman filter. Each of the matched filters is based on a hypothesized behavior of the residual, such as a sudden additive bias, a slowly increasing bias, an increase in measurement noise covariance, or no change in the residual. The hypothesized residual behaviors are based on the designer's expectations that a certain failure will cause this behavior in the residual, and they are often rather ad hoc and simple in practice. This interpretation of our results holds that the linear transforms operating on the output of a single Kalman filter (as seen in Figure 6, 25, and 27) are equivalent to the best possible "matched filters" of the GLR structure. This allows the designer to postulate a certain failure model and compute the linear transform for that model using the postulated differences between the fully functional model and the hypothesized failure model. The linear transform is equivalent to the best possible GLR matched

filter for that failure model. The interpretation establishes an equivalence class relationship between MMAE failure identification and the best-possible GLR failure detection and isolation systems.

#### 5.4 Residual Correlation Kalman Filter Bank

We developed a new kind of filter bank structure in Section 3.2.6, in which an estimation of the spectral content of the residuals (or equivalent residuals) is used by the hypothesis testing algorithm. The actuator failure identification performance of this Residual Correlation Kalman Filter Bank (RCKFB) was tested and the results were presented in Section 4.4.2, and a comparison of this performance against other techniques was made in Section 4.4.4. In Section 3.2.4, we showed that, when an actuator failure occurs, elements of the control input are added to the residual. If the control input elements are sinusoids, then the residual includes a sinusoid, with the same frequency as the control input elements. Therefore, we can use the spectral content of the residual, at the input frequencies, to indicated the presence of an actuator failure. We found that, when the control input is fairly strong, this technique takes longer than the standard MMAE to make the correct failure identification, due to the time used to collect a sufficient number of samples to generate an adequate spectral estimate. However, this technique provides failure identification performance at small input levels where the standard MMAE no longer functions. This technique looks quite promising as a method to provide good failure identification with subliminal input dithers, which could be used in conjunction with the standard MMAE structure to provide failure identification performance at low input levels and to corroborate the failure identification of the SKFB-SHTA structure at higher input levels.

## 5.5 Neyman-Pearson Hypothesis Testing Algorithm

We developed a Neyman-Pearson Hypothesis Testing Algorithm (NPHTA) in Section 3.3.2. We first extended the standard development for a scalar process, to a multidimensional process, and then extended that from a single time sample hypothesis test, to multiple time sample hypothesis testing. Sequential Probability Ratio Tests (SPRT) [34] have been shown to be superior to the Neyman-Pearson most powerful tests when operating on sequential data, because the SPRT allows variable sample size and makes a decision when and only when sufficient information is obtained. We extended the NPHTA to multiple time samples by incorporating a triggering methodology, and so our algorithm should share many of the attributes of the SPRT algorithm.

We showed that, by using the NPHTA, we could perform multiple hypothesis tests on a single Kalman filter residual, instead of having to form a residual for each hypothesis. Thus, we would use a Kalman filter "bank" of a single Kalman filter, followed by the NPHTA. This structure was implemented and tested, and the results were presented in Section 4.4.3. The actuator failure identification performance was compared to other MMAE structures in Section 4.4.4. We found that this technique was significantly faster than the standard MMAE, when the input strength was large enough to provide good failure identification performance. When the input strength dropped below this minimum level that was needed for good failure identification, the performance of this technique dropped off dramatically. The performance for sensor failures was found to be quite good for sensors that do not have strong measurement noise, but the false alarm rate was rather high for sensors that have strong measurement noise. The performance of the NPHTA, along with the single Kalman filter "bank" structure, argue strongly for implementing this structure for applications for which the input strength can be above the minimum level needed for good failure identification. However, if the input levels are below this minimum level, the RCKFB structure would need to be

implemented. Thus, we foresee having a two-structure MMAE algorithm: a RCKFB-based MMAE for low input strengths, and an NPHTA for high input strengths. A three-structure MMAE may be desired, where the third structure is the standard MMAE, to corroborate the results from the other two structures. All of these structures could use the equivalent Kalman filter bank composed of a single Kalman filter with a series of linear transforms to produce the equivalent residual of a fully implemented SKFB.

## 5.6 Recommendations for Future Research

This investigation opens up several possible avenues of further research. First, and foremost, would be combining the RCKFB with the NPHTA. This has the potential to provide the excellent failure identification performance of the NPHTA at high input strength, with the good failure identification performance of the RCKFB at low input strength. Along with this, the performance of the RCKFB for various values of  $N$ , the number of data points used by the spectral estimator, needs to be characterized. Further, the use of other methods of spectral estimation, such as the Blackman-Tukey, model-based methods, and Maximum Entropy Methods (MEM), needs to be investigated. The failure identification performance of these other techniques should be evaluated for each of the different methods, and a comparison made with the periodogram that was used for this research. Also, the performance of the NPHTA needs to be explored for various detection and false alarm probability settings, and a method of determining the NPHTA sensor failure performance needs to be developed.

Several of the MMAE structures that were developed in the research, show great promise in reducing the computational loading of the MMAE: specifically, the single residual Kalman filter bank, developed in Section 3.2.5, and the single Kalman filter bank used by the NPHTA. The computational savings attained by these structures need to be determined, for various applications,

by further research. It is particularly attractive for actuator failure identification because of the sparseness of the system input matrix that is used by the linear transformations that replace the many Kalman filters of a standard MMAE algorithm. Implementing this structure for sensor failure identification may produce computational savings if the difference between the Kalman filter gain matrices results in a sparse matrix. Also, the computational cost of implementing the RCKFB using Eq (71), in which the full estimation of the autocorrelation matrix is used, needs to be compared to the savings attained by implementing Eq (72), where only the diagonal terms of the estimated autocorrelation matrix are used, thus ignoring the cross-correlation terms. This cost savings needs to be determined and compared to any reduction in failure identification performance caused by ignoring the cross-correlation terms.

The variation in failure identification times caused by different failure injection times needs to be examined. The design of "balanced" actuator inputs, in which the input for one actuator has an equivalent effect on the likelihood quotient as the input for another actuator, needs to be a part of this examination. This would help determine the appropriate mix of inputs to provide good failure identification of all pertinent failure modes.

This suggests an alternative structure to the RCKFB, one that uses the spectral content of the scalar likelihood quotients from the various Kalman filters, rather than the spectral content of the residual vector. This structure would exploit the cross-coupling terms that were discarded when we implemented the Fourier transform-based spectral estimator. Thus, this structure would use the simpler and less computationally intensive Fourier transform-based estimator, as in Eq (72), but still include all of the cross-terms that are in the periodogram-based spectral estimator of Eq (71). The development of the algorithm shows great promise for improving the RCKFB failure identification performance.

Future research should aim at an information-theoretic development of a class of algorithms that fully (or partially) exploits the information content in the residuals. The algorithm structures that were developed in this research would be examples of elements of such a class of algorithms.

A fuller, more rigorous investigation of the interrelationship between the equivalent-residual MMAE-like structures and the GLR structure that uses the best-possible matched filters merits being accomplished. The linear transforms of this research are the best matched filters in the sense that they truly represent the time history of the residuals caused by a physically motivated failure model.

## Bibliography

1. Athans, M. *et al.* "The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part I: Equilibrium Flight," *IEEE Transactions on Automatic Control*, AC-22(5): 768 - 780, (October 1977).
2. Bendat, J. S., and A. G. Peirsol. *Random Data: Analysis and Measurement Procedures*. New York: Wiley, 1971.
3. Blakelock, John H. *Automatic Control of Aircraft and Missiles*. New York: John Wiley & Sons, Inc., 1991
4. Chamberlain, J. *The Principles of Interferometric Spectroscopy*. New York: Wiley, 1971.
5. Lashlee, R. W., Jr., and P. S. Maybeck. "Spacestructure Control Using Moving Bank Multiple Model Adaptive Estimation," *Proceedings of the IEEE Conference on Decision and Control*, San Antonio, Texas, (December 1988).
6. Chang, C. B., and M. Athans. "Hypothesis Testing and State Estimation for Discrete Systems with Finite-Valued Switching Parameters," MIT Electronic Systems Laboratory publication ESL-P-758, Lexington, MA, (June 1977).
7. Chang, C. B. "State Estimation for Discrete Systems with Switching Parameters," *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(4): 418 - 425, (May 1978).
8. Chang, C. B., and J. A. Tabaczynski. "Application of State Estimation to Target Tracking," *IEEE Transactions on Automatic Control*, AC-29(2): 98 - 109, (February 1984).
9. Dugapta, S., and L. C. Westphal. "Convergence of Partitioned Adaptive Filters for Systems with Unknown Biases," *IEEE Transactions on Automatic Control*, AC-28(5): 614 - 615, (May 1983).
10. Eckert, S. J., and K. A. Loparo. "An Application of Nonlinear Filtering to Instrument Failure Detection in a Pressurized Water Reactor," *Nuclear Technology*, 74: 139 - 150, (August 1986).
11. Eide, Peter K. *Implementation and Demonstration of a Multiple Model Adaptive Estimation Failure Detection System for the F-16*. MS Thesis, AFIT/GE/ENG/94D-06. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1994.
12. Eide, P. K. and P. S. Maybeck. "An MMAE Failure Detection System for the F-16," *IEEE Transactions of Aerospace and Electronic Systems*, AES-32(3): 1125-1136, (July 1996).

13. Filios, P. G. *Moving-Bank Multiple Model Adaptive Algorithms Applied to Flexible Spacecraft Control*. MS Thesis, AFIT/GE/ENG/85D-14. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1985.
14. Flynn, 2Lt Patrick M. *Alternative Dynamics Models and Multiple Model Filtering for a Short Range Tracker*. MS Thesis, AFIT/GE/EE/81D-21. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981.
15. Gholson, N. H., and R. L. Moose. "Maneuvering Target Tracking Using Adaptive State Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, AES-13(3): 310 - 317, (May 1977).
16. Greene, C. S., and A. S. Willsky. "An Analysis of the Multiple Model Adaptive Control Algorithm," *Proceedings of the IEEE Conference on Decision and Control*, Albuquerque, New Mexico: pp. 1142 - 1145 ( Dec. 1980).
17. Gustafson, D. E., A. S. Willsky, and J.-Y. Wang. "Final Report: Cardiac Arrhythmia Detection and Classification Through Signal Analysis," The Charles Stark Draper Laboratory, Report Number R-920, Cambridge, MA, July 1975.
18. Jain, A. K. "Advances in Mathematical Modeling for Image Processing," *Proceedings of the IEEE*, 69: 502-528, (May 1981).
19. Hanlon, P. D. *Failure Identification using Multiple Model Adaptive Estimation for the LAMBDA Flight Vehicle*. MS Thesis, AFIT/GE/ENG/92D-19. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.
20. Hawkes, R. M., and J. B. Moore. "Performance Bounds for Adaptive Estimation," *Proceedings of the IEEE*, 64: 1143 - 1150, (August 1976).
21. Hentz, K. P. *Feasibility Analysis of Moving Bank Multiple Model Adaptive Estimation and Control Algorithms*. MS Thesis, AFIT/GE/ENG/84D-32. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1984.
22. Hostetler, L. D., and R. D. Andreas. "Nonlinear Kalman Filtering Techniques for Terrain-Aided Navigation," *IEEE Transactions on Automatic Control*, AC-28(3): 315 - 323, (March 1983).
23. Karnick, D. A. *Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Space-Structure Control*. MS Thesis, AFIT/EN/ENG/86D-41. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.
24. Karnick, D. A., and P. S. Maybeck. "Moving Bank Multiple Model Adaptive Estimation Applied to Flexible Spacestructure Control," *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, California: pp 1249 - 1257, (December 1987).

25. Kay, S. M. *Autoregressive Spectral Analysis of Narrowband Processes in White Noise with Application to Sonar Signals*. PhD Dissertation, Georgia Institute of Technology, 1980.
26. Kay, Steven M. *Modern Spectral Estimation*. Englewood Cliffs, NJ: Prentice Hall, 1988.
27. Korn, J., and L. Beean. "Application of Multiple Model Adaptive Estimation Algorithms to Maneuver Detection and Estimation," Technical Report TR-152, Alphatech, Inc., Burlington, Mass., June 1983.
28. Lainiotis, D. G. "Partitioning: A Unifying Framework for Adaptive Systems, I: Estimation," *Proceedings of the IEEE*, 64: 1126 - 1142, (August 1976).
29. Lainiotis, D. G. "Partitioning: A Unifying Framework for Adaptive Systems, II: Control," *Proceedings of the IEEE*, 64: 1182 - 1197, (August 1976).
30. Lander, T. E., and R. T. Lacoss. "Some Geophysical Applications of Autoregressive Spectral Estimates," *IEEE Transactions on Geoscience and Electronics*, GE-15: 26-32, January 1977.
31. Lane, David W. *Multiple Model Adaptive Estimation Applied to the LAMBDA URV for Failure Detection and Identification*. MS Thesis, AFIT/GE/ENG/93D-23. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993.
32. Lashlee, R. W., Jr. *Moving-Bank Multiple Model Adaptive Estimation Applied to Flexible Space-Structure Control*. MS Thesis, AFIT/GE/ENG/87D-36. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1987.
33. Lashlee, R. W., Jr., and P. S. Maybeck. "Spacestructure Control Using Moving Bank Multiple Model Adaptive Estimation," *Proceedings of the IEEE Conference on Decision and Control*, San Antonio, Texas, (December 1988).
34. Li, X. R., and Y. Bar-Shalom. "Multiple-Model Estimation with Variable Structure," *IEEE Transactions on Automatic Control*, AC-41(4): 482 - 439, (April 1996).
35. Magill, D. T. "Optimal Adaptive Estimation of Sample Stochastic Processes," *IEEE Transactions on Automatic Control*, AC-10(4): 434 - 439, (October 1965).
36. Makhoul, J., R. Viswanathan, L. Cosell, and W. Russell. "Natural Communication with Computers: Speech Compression Research at BBN," BBN Report 2976 II, Bolt, Beranek, and Newman, Inc., Cambridge, Mass., 1974
37. Maybeck, P. S. *Stochastic Models, Estimation, and Control*, Vol. 1. New York: Academic Press, 1979. Republished Arlington, VA: Navtech, 1994.

38. Maybeck, P. S. *Stochastic Models, Estimation, and Control*, Vol. 2. New York: Academic Press, 1979. Republished Arlington, VA: Navtech, 1994.
39. Maybeck, Peter S. Professor, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH. Personal Interview. 6 September, 1996.
40. Maybeck, P. S., and K. P. Hentz. "Investigation of Moving-Bank Multiple Model Adaptive Algorithms," *AIAA Journal of Guidance, Control, and Dynamics*, Volume 10, Number 1: pp. 90 - 96, (January-February 1987).
41. Maybeck, P.S., and P. D. Hanlon. "Performance Enhancement of a Multiple Model Adaptive Estimator," *IEEE Transactions on Aerospace and Electronic Systems*, AES-31(4): 1240 - 1253, (October 1995).
42. Maybeck, P. S., and D. L. Pogoda. "Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures," *Proceedings of the IEEE Conference on Decision and Control*, Tampa, Florida: pp. 1566 - 1572, (December 1989).
43. Maybeck, P. S., and R. D. Stevens. "Reconfigurable Flight Control via Multiple Model Adaptive Control Methods," *IEEE Transactions on Aerospace and Electronic Systems*, AES-27(3): 470 - 480, (May 1991).
44. Maybeck, P. S., and R. I. Suizu. "Adaptive Tracker Field-of-View Variation Via Multiple Model Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(4): 529 - 539, (July 1985).
45. Maybeck, P. S., and W. L. Zicker. "MMAE-Based Control with Space-Time Point Process Observations," *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3): 292 - 300, (May 1985).
46. Mealy, G. L., and W. Tang. "Application of Multiple Model Estimation to a Recursive Terrain Height Correlation System," *IEEE Transactions on Automatic Control*, AC-28(3): 323 - 331, (March 1983).
47. Menke, Timothy E. *Multiple Model Adaptive Estimation Applied to the VISTA F-16 with Actuator and Sensor Failures*. MS Thesis, AFIT/GA/ENG/92J-01. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1992.
48. Menke, Timothy E., and P. S. Maybeck. "Multiple Model Adaptive Estimation Applied to the VISTA F-16 Flight Control System with Actuator and Sensor Failures," *Proceedings of the IEEE of the National Aerospace and Electronics Conference (NAECON)*, Dayton, Ohio: pp 441 - 448 (May 1992).
49. Menke, Timothy E., and P. S. Maybeck. "Sensor/Actuator Failure Detection in the VISTA F-16 by Multiple Model Adaptive Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, AES-31(4): 1218-1229 (October 1995).

50. Moose, R. L. "An Adaptive State Estimation Solution to the Maneuvering Target Problem," *IEEE Transactions on Automatic Control*, AC-20(3): 359 - 362 (June 1975).
51. Moose, R. L., H. F. Van Landingham, and D. H. McCabe. "Modeling and Estimation for Tracking Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, AES-15(3): 448 - 456, (May 1979).
52. Pogoda, Capt Donald L. *Multiple Model Adaptive Controller for the STOL F-15 with Sensor/Actuator Failures*. MS Thesis, AFIT/GE/ENG/88D-37. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.
53. Scharf, L. L. *Statistical Signal Processing*. Reading, Massachusetts: Addison-Wesley, 1991.
54. Schore, M. R. *Robustness of a Moving-Bank Multiple Model Adaptive Controller for a Large Space Structure*. MS Thesis, AFIT/GE/ENG/89D-46. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.
55. Sheldon, Capt Stuart N. *An Optimal Parameter Discretization Strategy for Multiple Model Adaptive Estimation and Control*. PhD Dissertation, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.
56. Sheldon, S. N., and P. S. Maybeck. "An Optimizing Design Strategy for Multiple Model Adaptive Estimation and Control," *IEEE Transactions on Automatic Control*, AC-38(4): 651 - 654, (April 1993).
57. Sims, C. S., and M. R. D'Mello. "Adaptive Deconvolution of Seismic Signals," *IEEE Transactions on Geoscience Electronics*, GE-16: 99 - 103, (April 1978).
58. Stepaniak, Michael J. *Multiple Model Adaptive Control of the VISTA F-16*. MS Thesis, AFIT/GE/ENG/95D-26. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1995.
59. Stepaniak, M. J. and P. S. Maybeck. "MMAE-Based Control Redistribution Applied to the VISTA F-16," to appear in the *Proceedings of the American Control Conference*, (June 1997).
60. Stevens, Capt Richard D. *Characterization of a Reconfigurable Multiple Model Adaptive Controller Using A STOL F-15 Model*. MS Thesis, AFIT/GE/ENG/89D-52. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1989.
61. Stratton, Capt Gregory L. *Actuator and Sensor Failure Identification Using a Multiple Model Adaptive Technique for the VISTA/F-16*. MS Thesis, AFIT/GE/ENG/91D-53. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991.

62. Swift, 1Lt Gerald A. *Model Identification and Control System Design for the LAMBDA Unmanned Research Vehicle*. MS Thesis, AFIT/GAE/ENY/91S-4. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1991.
63. Tenney, R. R., R. S. Hebbert, and N. R. Sandell, Jr. "A Tracking Filter for Maneuvering Sources," *IEEE Transactions on Automatic Control*, AC-22(2): 246 - 261, (March 1977).
64. Thorp, J. S. "Optimal Tracking of Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, AES-9(4): 512 - 519, (July 1973).
65. Tobin, D. M., and P. S. Maybeck. "Substantial Enhancements to a Multiple Model Adaptive Estimator for Target Image Tracking," *Proceedings of the IEEE Conference on Decision and Control*, Los Angeles, CA: 2002 - 2011, (December 1987).
66. Vasquez, Juan R. *Detection of Spoofing, Jamming, or Failure of a Global Positioning System (GPS)*. MS Thesis, AFIT/GE/ENG/92D-37. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.
67. Wernecke, S. "Two-Dimensional Maximum Entropy Reconstruction of Radio Brightness," *Radio Science*, 12: 831-844, (September-October 1977).
68. Willsky, A. S. "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, 12: 601 - 611, (December 1976).
69. Willsky, A. S. *et al.* "Dynamic Model-Based Techniques for the Detection of Incidents on Freeways," *IEEE Transactions on Automatic Controls*, AC-25(3): 347 - 359, (June 1980).

## Vita

Captain Peter D. Hanlon [REDACTED] He served at numerous Air Force assignments as an active duty dependent, settling, finally, at Willingboro, New Jersey where he graduated from Kennedy High School in 1974. He attended Burlington County College and Rutgers University, Cook College where he studied Meteorology. He enlisted in the Air Force in 1979, trained as a voice processing specialist, and was assigned to Athens, Greece in 1981 where he flew as a crew member on RC-135's. He was accepted into the Airman's Education and Commissioning Program (AECMP) in 1983 and was assigned to the University of Central Florida, Orlando, Florida until his graduation in 1985 with a Bachelor of Science in Engineering. After his commissioning, he was assigned to the Geophysics Laboratory at Hansom Air Force Base, Massachusetts, where he served as either program manager or project engineer for several space shuttle, sounding rocket, and high altitude balloon research projects until 1989. He was miserably assigned to the staff of the Aeronautical Systems Division, Wright-Patterson Air Force Base until he escaped to the School of Engineering, Air Force Institute of Technology in 1991. He earned a Master of Science in Electrical Engineering in December 1992, and continued on at the Air Force Institute of Technology as a Doctorate of Philosophy student.

[REDACTED] Present Address: [REDACTED]  
[REDACTED]

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Doctoral Dissertation		
4. TITLE AND SUBTITLE PRACTICAL IMPLEMENTATION OF MULTIPLE MODEL ADAPTIVE ESTIMATION USING NEYMAN-PEARSON HYPOTHESIS TESTING AND SPECTRAL ESTIMATION TOOLS		5. FUNDING NUMBERS		
6. AUTHOR(S)  Peter D. Hanlon, Capt, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Air Force Institute of Technology 2750 P Street, WPAFB OH 45433-7765		8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/DS/ENG/96-07		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Lt Christina L. Osmon WL/FIGS-2, 2210 8th Street, Suite 11 WPAFB OH 45433		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This study investigates and develops various modifications to the Multiple Model Adaptive Estimation (MMAE) algorithm. The standard MMAE uses a bank of Kalman filters, each based on a different model of the system. Each of the filters predict the system response, based on its system model, to a given input and form the residual difference between the prediction and sensor measurements of the system response. Model differences in the input matrix, output matrix, and state transition matrix, which respectively correspond to an actuator failure, sensor failure, and an incorrectly modeled flight condition for a flight control failure application, were investigated in this research. An alternative filter bank structure is developed that uses a linear transform on the residual from a single Kalman filter to produce the equivalent residuals of the other Kalman filters in the standard MMAE. A Neyman-Pearson based hypothesis testing algorithm is developed that results in significant improvement in failure detection performance when compared to the standard hypothesis testing algorithm. Hypothesis testing using spectral estimation techniques is also developed which provides superior failure identification performance at extremely small input levels.				
14. SUBJECT TERMS  Kalman Filter, Multiple Model Adaptation, Failure Identification			15. NUMBER OF PAGES 197	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

**Block 1. Agency Use Only (Leave blank).**

**Block 2. Report Date.** Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

**Block 3. Type of Report and Dates Covered.** State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4. Title and Subtitle.** A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5. Funding Numbers.** To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

<b>C</b> - Contract	<b>PR</b> - Project
<b>G</b> - Grant	<b>TA</b> - Task
<b>PE</b> - Program Element	<b>WU</b> - Work Unit Accession No.

**Block 6. Author(s).** Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7. Performing Organization Name(s) and Address(es).** Self-explanatory.

**Block 8. Performing Organization Report Number.** Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es).** Self-explanatory.

**Block 10. Sponsoring/Monitoring Agency Report Number.** (If known)

**Block 11. Supplementary Notes.** Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

**Block 12a. Distribution/Availability Statement.** Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

**DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."

**DOE** - See authorities.

**NASA** - See Handbook NHB 2200.2.

**NTIS** - Leave blank.

**Block 12b. Distribution Code.**

**DOD** - Leave blank.

**DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

**NASA** - Leave blank.

**NTIS** - Leave blank.

**Block 13. Abstract.** Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

**Block 14. Subject Terms.** Keywords or phrases identifying major subjects in the report.

**Block 15. Number of Pages.** Enter the total number of pages.

**Block 16. Price Code.** Enter appropriate price code (*NTIS only*).

**Blocks 17. - 19. Security Classifications.** Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

**Block 20. Limitation of Abstract.** This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.